



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

DEPARTMENT OF CONTROL AND INSTRUMENTATION

FLEXIBILNÍ BEZDRÁTOVÝ SYSTÉM PRO MĚŘENÍ CO₂ V BUDOVĚ

INDOOR FLEXIBLE WIRELESS CO₂ MEASURE SYSTEM

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Vít Válek

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Jakub Arm, Ph.D.

BRNO 2021

Diplomová práce

magisterský navazující studijní program **Kybernetika, automatizace a měření**

Ústav automatizace a měřicí techniky

Student: Bc. Vít Válek

ID: 195457

Ročník: 2

Akademický rok: 2020/21

NÁZEV TÉMATU:

Flexibilní bezdrátový systém pro měření CO₂ v budově

POKYNY PRO VYPRACOVÁNÍ:

Práce se zabývá návrhem a realizací měřicího systému skládajícího se z více zařízení, které měří koncentraci CO₂ v dané lokalitě a které komunikují s nadřazeným koncentrátorem dat pomocí bezdrátové sítě typu mesh (Bluetooth 5.0). K vytvoření je možné využít vhodných vývojových modulů. Důraz je kladen na podporu mesh funkce (přenos zpráv také pomocí dalších účastníků sítě), zpřístupnění dat z CO₂ měřicího zařízení přímo (pomocí smartphone) a podporu flexibilní (plug-in) změny topologie sítě, tj. přidání/odebrání zařízení přímo za běhu bez nutnosti restartu celého systému.

- 1) Seznamte se standardem Bluetooth 5.0.
- 2) Navrhněte architekturu měřicího systému.
- 3) Realizujte systém pomocí vhodných komponent.
- 4) Implementujte SW vybavení.
- 5) Otestujte funkčnost (mesh, flexibilní plug-in a přímý přístup k měřicímu zařízení).
- 6) Změřte parametry vytvořeného systému (spolehlivost, rychlost propagace datové zprávy).

DOPORUČENÁ LITERATURA:

AFANEH, M. Bluetooth 5 & Bluetooth Low Energy, a developer's guide. NovelBits, 2020.

Termín zadání: 8.2.2021

Termín odevzdání: 17.5.2021

Vedoucí práce: Ing. Jakub Arm, Ph.D.

doc. Ing. Petr Fiedler, Ph.D.
předseda rady studijního programu

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Monitorování koncentrace oxidu uhličitého v budově se provádí z několika důvodů. Jedním z nich je zajištění podmínek hygieny. S nástupem Bluetooth 5.0 přišla podpora síťové technologie mesh, která je definována standardem Bluetooth Mesh. Implementací tohoto standardu můžeme vytvořit rozsáhlou síť zařízení monitorujících koncentraci oxidu uhličitého v budově. Na základě monitorované koncentrace můžeme řídit klimatizaci a větrání prostorů, čímž zajistíme splnění hygienických podmínek. Díky kompatibilitě Bluetooth Mesh s Bluetooth Low Energy je možné přistoupit k jednotlivým uzlům např. z mobilního telefonu. Cílem této práce je navrhnout a realizovat měřicí systém, jehož prvky spolu budou komunikovat pomocí bezdrátové technologie Bluetooth Mesh.

KLÍČOVÁ SLOVA

Bluetooth Mesh, koncentrátor, perioda publikování, uzel, senzor CO₂, MCU

ABSTRACT

Monitoring of the carbon dioxide concentration in the building is carried out for several reasons. One is to ensure hygiene conditions. With the advent of Bluetooth 5.0 came the support of mesh network technology, which is defined by the Bluetooth Mesh standard. By implementing this standard, we can create an extensive network of devices monitoring the concentration of carbon dioxide in the building. Based on the monitored concentration, we can control the air conditioning and ventilation of the spaces, ensuring that the hygiene conditions are met. Thanks to the compatibility of Bluetooth Mesh with Bluetooth Low Energy, it is possible to access individual nodes, e.g. from a mobile phone. The aim of this work is to design and implement a measuring system whose elements will communicate with each other using Bluetooth Mesh wireless technology.

KEYWORDS

Bluetooth Mesh, concentrator, publish period, node, sensor CO₂, MCU

VÁLEK, Vít. *Flexibilní bezdrátový systém pro měření CO₂ v budově*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky, 2021, 82 s. Diplomová práce. Vedoucí práce: Ing. Jakub Arm, PhD.

Prohlášení autora o původnosti díla

Jméno a příjmení autora: Bc. Vít Válek
VUT ID autora: 195457
Typ práce: Diplomová práce
Akademický rok: 2020/21
Téma závěrečné práce: Flexibilní bezdrátový systém pro měření CO₂ v budově

Prohlašuji, že svou závěrečnou práci jsem vypracoval samostatně pod vedením vedoucí/ho závěrečné práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené závěrečné práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno
.....
podpis autora*

*Autor podepisuje pouze v tištěné verzi.

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu diplomové práce panu Ing. Jakubovi Armovi, Ph.D. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé diplomové práce.

Obsah

Úvod	14
1 Komunikační technologie Bluetooth	15
1.1 Bluetooth Classic	15
1.2 Bluetooth Low Energy	15
1.3 Bluetooth 5.0	16
1.4 Přenosové režimy Bluetooth 5.0	16
1.5 Architektura Bluetooth 5.0	17
1.5.1 Fyzická vrstva (Physical Layer - PHY)	18
1.5.2 Linková vrstva (Link Layer)	18
1.5.3 Režim pro přímé testování (Direct Test Mode)	20
1.5.4 Rozhraní hostitel-ovladač (HCI)	20
1.5.5 Generic Access Profile (GAP)	21
1.5.6 Logical Link Control & Adaption Protocol (L2CAP)	25
1.5.7 Attribute Protocol (ATT)	25
1.5.8 Security Manager	26
1.5.9 Generic Attribute Profile (GATT)	26
2 Bluetooth mesh	28
2.1 Architektura Bluetooth mesh	28
2.2 Prvky a vlastnosti Bluetooth mesh	29
2.2.1 Uzly, elementy, modely	29
2.2.2 Stavby	31
2.2.3 Zprávy	31
2.2.4 Adresy	31
2.2.5 Komunikace v rámci mesh	32
2.3 Zabezpečení v Bluetooth mesh	33
2.4 Přidání zařízení do mesh	33
3 Měření oxidu uhličitého	35
3.1 Snímače oxidu uhličitého	35
3.2 Vybrané snímače oxidu uhličitého	36
4 Použité MCU	38
4.1 STMicroelectronics	38
4.2 Nordic Semiconductor	39

5	Návrh měřicího systému	41
5.1	Podmínky a prostory pro měřicí systém	41
5.2	Architektura systému z pohledu Bluetooth mesh	43
5.2.1	Modely a konfigurace adres	43
5.3	Koncentrátor	45
5.4	Měřicí bod	46
6	Realizace měřicího systému	47
6.1	Koncentrátor	47
6.1.1	Přidání zařízení do sítě	48
6.1.2	Odebrání zařízení ze sítě	50
6.1.3	Přidání a odebrání publish/subscribe adres	52
6.1.4	Příjem dat ze senzorů	52
6.1.5	Logování měřených hodnot	53
6.1.6	Měření latence propagace datové zprávy	53
6.1.7	Ovládání aplikace	55
6.2	Měřicí bod	56
6.3	Začlenění nRF52840 USB Dongle	57
6.4	Odebrání zařízení ze sítě pomocí uživatelských tlačítek	57
6.5	Komplikace, které se nepovedlo vyřešit	58
7	Naměřené parametry systému	60
7.1	Úspěšnost přenosu a latence datové zprávy	60
7.1.1	Konfigurace 1	61
7.1.2	Konfigurace 2	63
7.1.3	Konfigurace 3	65
7.1.4	Konfigurace 4	66
7.2	Průměrný odběr proudu zařízení	68
7.2.1	Proudový odběr koncentrátoru	68
7.2.2	Proudový odběr Nucleo68 s MH-Z14A	71
7.2.3	Proudový odběr WB55 USB Dongle s MH-Z19B	72
7.3	Měření parametrů LPN	73
	Závěr	76
	Literatura	77
	Seznam příloh	81

A Projekty a zdrojové kódy	82
A.1 Koncentrátor	82
A.2 Nucleo68 - Sensor server	82
A.3 WB55 USB Dongle - Sensor server	82

Seznam obrázků

1.1	Konfigurace a kompatibilita Bluetooth zařízení	16
1.2	Architektura Bluetooth Low Energy [4]	17
1.3	Frekvenční spektrum Bluetooth Low Energy [3]	18
1.4	Stavový automat linkové vrstvy	19
1.5	Formát paketu Bluetooth 5.0	20
1.6	Vysílání a skenování propagačních zpráv [6]	22
1.7	Interval připojení a zpoždění slave [5]	23
1.8	PDU propagačních zpráv	24
1.9	Užitečná zpráva v PDU [2]	24
1.10	Formát datové části v PDU [2]	25
1.11	Formát atributu	26
1.12	GATT: Profily, služby, charakteristiky[2]	27
2.1	Architektura Bluetooth mesh[19]	28
2.2	Spolupráce mezi LPN a friend node	30
2.3	Ukázka modelu publish-subscribe [1]	33
2.4	Přidání neprovozovaného zařízení do sítě mesh [19]	34
3.1	NDIR snímač CO ₂ [13]	36
4.1	Programování P-Nucleo-WB55 USB Dongle	39
4.2	Mód DFU - výběr COM portu	40
5.1	Půdorys 3.NP	41
5.2	Půdorys 2.NP	42
5.3	Půdorys 1.NP	42
5.4	Mesh síť měřicího systému	43
5.5	Použité modely a konfigurace adres pro publikování a odebírání zpráv	44
5.6	Blokové schéma koncentrátoru	45
5.7	Koncentrátor (vlevo) a měřicí body	46
6.1	Obsah NVM při 2 uzlech v síti mesh (před úpravou)	49
6.2	Obsah NVM při 2 uzlech v síti mesh (po úpravě)	50
6.3	Odebrání zařízení ze sítě mesh	51
6.4	Příjem a ukládání měřených dat	52
6.5	Logování měřených dat	54
6.6	Měření latence propagace datové zprávy	54
6.7	Odebrání nRF52840 USB Dongle pomocí SW1	58
7.1	Úspěšnost přenosu datové zprávy, konfigurace 1	62
7.2	Latence zprávy, konfigurace 1	62
7.3	Úspěšnost přenosu datové zprávy, konfigurace 2	64
7.4	Latence zprávy, konfigurace 2	64

7.5	Úspěšnost přenosu datové zprávy, konfigurace 3	65
7.6	Latence zprávy, konfigurace 3	66
7.7	Úspěšnost přenosu datové zprávy, konfigurace 4	67
7.8	Latence zprávy, konfigurace 4	68
7.9	Průběh proudu - koncentrátor	69
7.10	Průběh proudu při zápisu na SD kartu	69
7.11	Přiblížení průběhu proudu koncentrátoru při zápisu na SD kartu . . .	70
7.12	Průběh proudu - Nucleo68 se snímačem MH-Z14A	71
7.13	Skoková změna proudu - Nucleo68 se snímačem MH-Z14A	71
7.14	Průběh proudu - WB55 USB Dongle se snímačem MH-Z19B	72
7.15	Skoková změna proudu - WB55 USB Dongle se snímačem MH-Z19B .	73
7.16	Průběh proudu LPN při požadavku na data a jejich čtení	75

Seznam tabulek

1.1	Přenosové režimy Bluetooth 5.0	17
2.1	Skupiny pro jednotlivé typy uzlů	32
3.1	Vliv CO ₂ na lidský organismus [12]	35
3.2	Parametry vybraných snímačů	37
5.1	Zapojení modulů koncentrátoru	45
7.1	Konfigurace měření	60
7.2	Přidělené adresy a funkce uzlů	60
7.3	Naměřené hodnoty konfigurace 1, payload 7 bytů	61
7.4	Naměřené hodnoty konfigurace 1, payload 10 bytů	61
7.5	Naměřené hodnoty konfigurace 2, payload 7 bytů	63
7.6	Naměřené hodnoty konfigurace 2, payload 10 bytů	63
7.7	Naměřené hodnoty konfigurace 3, payload 7 bytů	65
7.8	Naměřené hodnoty konfigurace 3, payload 10 bytů	65
7.9	Naměřené hodnoty konfigurace 4, payload 7 bytů	66
7.10	Naměřené hodnoty konfigurace 4, payload 10 bytů	67
7.11	Odběr proudu koncentrátorem - rozdělení na úseky	70
7.12	Odběr proudu - Nucleo68 se snímačem MH-Z14A - rozdělení na úseky	72
7.13	Odběr proudu - WB55 USB Dongle se snímačem MH-Z19B - rozdělení na úseky	73
7.14	Úspěšnost doručení zprávy z LPN (konfigurace 1)	74
7.15	Úspěšnost doručení zprávy z LPN (konfigurace 2)	74
7.16	Měřený odběr proudu ve spánkovém režimu (WB55 USB Dongle) . .	74

Seznam výpisů

6.1	Konfigurace koncentrátoru	47
6.2	Inicializace I2C, SPI, TIM2 a GPIO	48
6.3	Inicializace parametrů komunikace	48
6.4	Změna velikosti stránky pro data provisionera	49
6.5	Definice struktury MeasDatat	53
6.6	Konfigurace měřicího bodu	56
6.7	Použití stavu Generic OnOff pro vzdálenou kalibraci	56

Úvod

Monitorování vnitřního klimatu budov může být žádoucí z různých důvodů. Jedním z monitorovaných parametrů je oxid uhličitý, což je plyn bez barvy a bez zápachu. Je přirozenou součástí okolního vzduchu. Zvýšená koncentrace oxidu uhličitého může mít pro člověka následky jako jsou např. únava, bolest hlavy, ztráta pozornosti. Ve špatně odvětrávaných prostorech nebo v prostorech, kde dochází ke shromáždění vyššího počtu osob, může být zvýšená koncentrace oxidu uhličitého kritický parametr.

Obsahem práce bude navrhnout a realizovat flexibilní bezdrátový systém měření koncentrací oxidu uhličitého. Samotnému návrhu a realizaci předchází seznámení se se standardy Bluetooth Low Energy a Bluetooth Mesh. Standard Bluetooth Mesh je založen na standardu Bluetooth 5.0 a definuje podporu síťové topologie mesh. Architektura navrženého měřicího systému bude založena právě na topologii mesh. Síť bude tvořena pomocí vývojových modulů od společností STMicroelectronics a Nordic Semiconductor a bude obsahovat několik uzlů se senzory měřícími koncentraci oxidu uhličitého v okolním prostředí. O správu sítě se bude starat uživatel. Bude moci přidávat a odebírat zařízení ze sítě, aniž by byl restartován celý systém. Dále bude možné dynamicky nastavovat adresy, odkud modely Bluetooth Mesh odebírají zprávy a kam zprávy publikují. V rámci mesh sítě by měl být umožněn přímý přístup k měřicímu zařízení např. přes mobilní telefon nebo tablet. Vytvořený systém nebude propojen s žádnou databází, a tak bude pro ukládání měřených hodnot koncentrace použita SD karta.

Celá práce je spíše zaměřena na standard Bluetooth Mesh a jeho implementaci, než na samotné měření oxidu uhličitého. Z pohledu Bluetooth Mesh prakticky nezáleží na snímané veličině, jelikož Bluetooth Mesh definuje modely, které lze použít pro širokou škálu snímaných veličin.

1 Komunikační technologie Bluetooth

Kapitola se bude zabývat standardizovanou bezdrátovou komunikací Bluetooth. Nejdříve budou popsány rozdíly mezi Bluetooth Classic a Bluetooth Low Energy. Poté se zaměříme na architekturu Bluetooth 5.0 a její jednotlivé vrstvy.

Bluetooth je otevřený standard definovaný v IEEE 802.15.1. Na vývoj standardu a licencování technologií dohlíží organizace Bluetooth SIG (Bluetooth Special Interest Group). Bluetooth umožňuje bezdrátovou komunikaci dvou a více zařízení. Technologie pracuje ve frekvenčním pásmu 2,4 GHz, které je označováno zkratkou ISM (Industrial, Scientific and Medical). Toto pásmo používají průmyslové, vědecké a zdravotnické obory. V případě použití homologovaného zařízení v tomto pásmu odpadá povinnost spjatá s licenčním poplatkem.

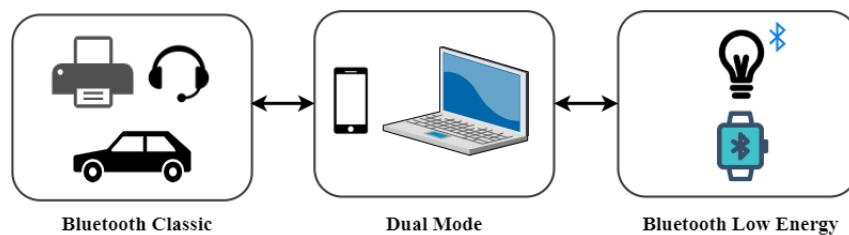
1.1 Bluetooth Classic

Termínem Bluetooth Classic jsou označována zařízení Bluetooth, které jsou ve specifikaci standardu označována zkratkou BR/EDR (The Basic Rate / Enhanced Data Rate). Bluetooth BR teoreticky podporuje datovou propustnost 1 Mbit/s. Propustnost Bluetooth Enhanced Data Rate může být až 3 Mbit/s. Propustnost bývá někdy vyjádřena v jednotce sym/s (symbol per second). 1 symbol reprezentuje 1 bit. Zařízení Bluetooth Classic můžeme nalézt např. v bezdrátových reproduktorech nebo v automobilech.

1.2 Bluetooth Low Energy

V roce 2010 byla společností Bluetooth SIG představena nová verze Bluetooth 4.0. Hlavním přínosem této verze bylo tzv. Bluetooth Low Energy (BLE), které je zachováno i ve všech pozdějších verzích. BLE cílí na nízkou spotřebu energie. Typický dosah je 10-30 m a propustnost 1 Mbit/s. Radiová komunikace probíhá na 40 kanálech s rozlišením 2 MHz. Pro výměnu dat slouží 37 kanálů. Zbývající 3 jsou vyhrazeny pro vysílání propagačních zpráv a vyhledávání okolních zařízení, což bude více popsáno v kapitole Linková vrstva (Link Layer). Zařízení s technologií BLE jsou např. chytré hodinky a náramky, meteostanice nebo různé prvky pro IoT.

Jak můžeme vidět na obr. 1.1, Bluetooth Classic a Bluetooth Low Energy nejsou vzájemně kompatibilní. Kompatibilitu zajistí zařízení, které má implementováno jak Bluetooth BR/EDR, tak i BLE. Jedná se o tzv. Dual Mode konfiguraci, se kterou se můžeme setkat v chytrých telefonech, v tabletech nebo v notebookech.



Obr. 1.1: Konfigurace a kompatibilita Bluetooth zařízení

1.3 Bluetooth 5.0

S příchodem Bluetooth 5.0 v roce 2016 nabývá standard na své popularitě. Datová propustnost může být až 2 Mbit/s. Je limitována rychlostí rádiového přenosu na fyzické vrstvě. Zvýšen byl i výstupní výkon, který zajistí větší dosah. Novinkou je také podpora síťové topologie mesh, která je velmi užitečná pro IoT.

1.4 Přenosové režimy Bluetooth 5.0

Datová propustnost a dosah závisí na použitém přenosovém režimu. Bluetooth 5.0 nabízí oproti svým předchůdcům 2 nové přenosové režimy, a to vysokorychlostní režim LE 2M a režim dlouhého dosahu LE Coded.

LE 1M

Tento režim je povinný ve všech zařízeních, které mají vyhovovat specifikacím standardu Bluetooth 5.0. Datová propustnost je 1 Mbit/s. Umožňuje detekovat chybu při přenosu díky cyklickému redundantnímu součtu (CRC z anglického Cyclic Redundancy Check). Oprava chyb není v tomto režimu možná.

LE 2M

Vysokorychlostní režim je oproti režimu LE 1M volitelný. Nabízí propustnost 2 Mbit/s. Pro detekci a opravu chyb platí stejné vlastnosti jako v případě LE 1M.

LE Coded

LE Coded je označení pro režim dlouhého dosahu. V tomto režimu rozlišujeme, kolik symbolů reprezentuje 1 bit. V módu S=2 je 1 bit reprezentován 2 symboly. Datová propustnost může být 500 Kbit/s na dvojnásobné vzdálenosti. V módu S=8 je analogicky 1 bit reprezentován 8 symboly. Dosah může být až čtyřnásobný při propustnosti 125 Kbit/s. Oba módy umožňují detekci chybného přenosu pomocí

CRC. Oproti LE 1M a LE 2M je možná i korekce chyby díky FEC (Forward Error Correction). Základní srovnání parametrů přenosových režimů je v tabulce 1.1.

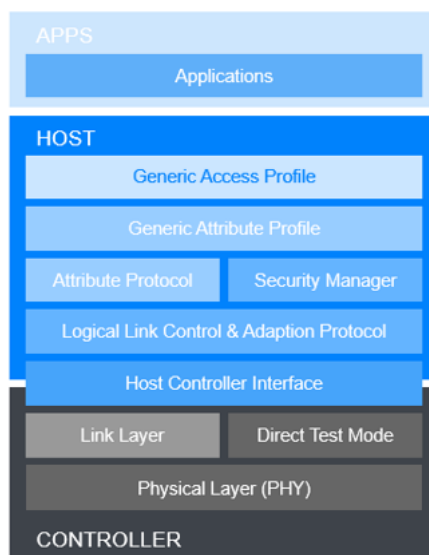
Tab. 1.1: Přenosové režimy Bluetooth 5.0

	LE 1M	LE 2M	LE Coded (S=2)	LE Coded (S=8)
Datová propustnost	1 Mbit/s	2 Mbit/s	500 Kbit/s	125 Kbit/s
Detekce chyby	CRC	CRC	CRC	CRC
Oprava chyby	-	-	FEC	FEC

1.5 Architektura Bluetooth 5.0

Jak můžeme vidět na obr.1.2, Bluetooth je tvořen třemi hlavními vrstvami: aplikační (Application), hostitel (Host), ovladač (Controller). Ty jsou dále děleny na podvrstvy a protokoly.

Nejvyšší vrstva je aplikační a není definována standardem. Může popisovat např. operace s daty, které byly přijaty. Jelikož její forma záleží na vývojáři, nebudeme se jí v této kapitole více zabývat.



Obr. 1.2: Architektura Bluetooth Low Energy [4]

1.5.1 Fyzická vrstva (Physical Layer - PHY)

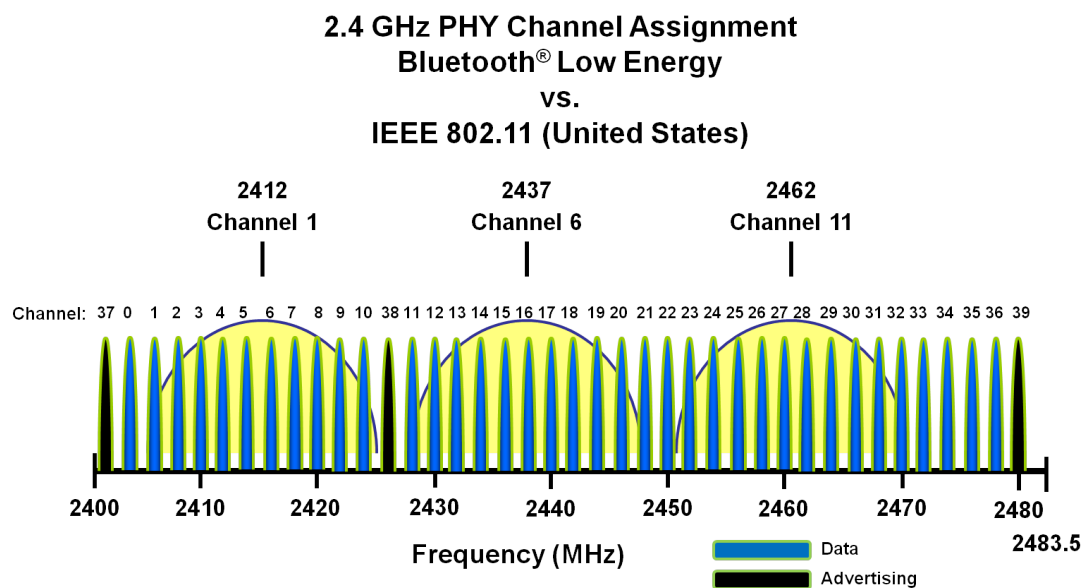
Již bylo zmíněno, že rádiové vysílání Bluetooth probíhá na 40 kanálech ve frekvenčním pásmu 2,4 GHz. Kanály 37, 38 a 39 slouží pro vysílání propagačních zpráv a vyhledávání okolních zařízení. Volba těchto tří kanálů má své opodstatnění. V pásmu 2,4 GHz pracuje i technologie Wi-Fi. Právě kanály 37 až 39 jsou oproti zbývajícím méně zarušeny kanály Wi-Fi, což můžeme vidět na obrázku níže. Na zbývajících kanálech probíhá výměna dat, která se netýká propagačních zpráv.

Počas spojení probíhá komunikace na více kanálech, a to z důvodu potlačení možných interferencí v pásmu 2,4 GHz. Pro výpočet nové frekvence, na které se bude komunikovat, je použita metoda FHSS (Frequency Hopping Spread Spectrum), jejíž vzorec je:

$$f_{k+1} = (f_k + hop) \bmod 37 \quad (1.1)$$

kde f_{k+1} je nová frekvence, f_k je aktuální frekvence a hodnota parametru hop je určena při navázání komunikace.

Dosah se pohybuje v rozmezí 60–240 m. Někteří výrobci komerčních Bluetooth modulů udávají dosah až 500 m [4].



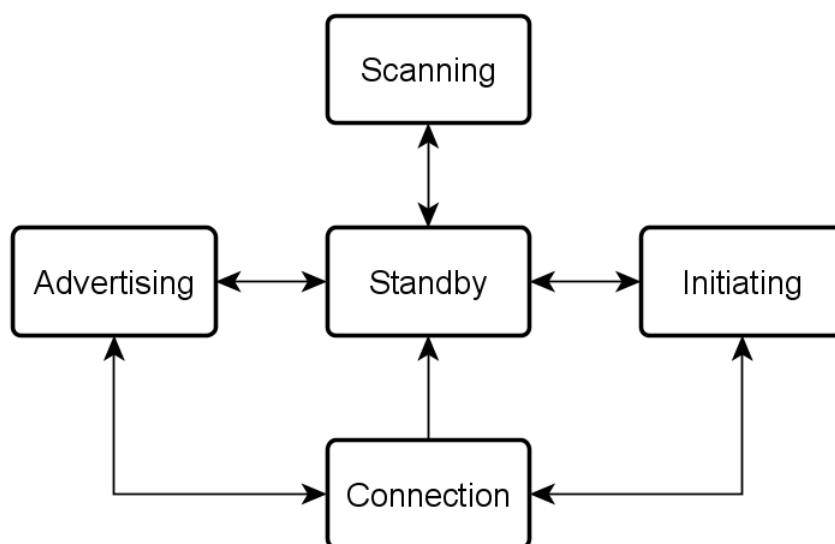
Obr. 1.3: Frekvenční spektrum Bluetooth Low Energy [3]

1.5.2 Linková vrstva (Link Layer)

Linková vrstva zajišťuje správu stavů zařízení, počítá cyklický redundantní součet (CRC), zodpovídá za generování náhodných čísel a v neposlední řadě obstarává

šifrování. Stavový automat na linkové vrstvě můžeme pospat pěti stavy:

- Standby – Zařízení nevysílá ani nepřijímá data.
- Advertising – Zařízení vysílá propagační zprávy na kanálech 37, 38 a 39, a zároveň poslouchá případnou odpověď na ně. Propagačními zprávami dává zařízení najevo svou přítomnost okolním zařízením.
- Scanning – Zařízení v tomto stavu vyhledává zařízení vysílající propagační zprávy. Jinými slovy, vyhledává zařízení ve stavu advertising.
- Initiating – Zařízení se v tomto stavu pokouší navázat spojení se zařízením, od kterého byly přijaty propagační zprávy.
- Connection – Stav, kdy je zařízení spojeno s dalším zařízením. V tomto stavu může mít zařízení dvě role:
 - Master – Stává se jím zařízení, jehož předchozí stav byl initiating.
 - Slave – Stává se jím zařízení, jehož předchozí stav byl advertising.



Obr. 1.4: Stavový automat linkové vrstvy

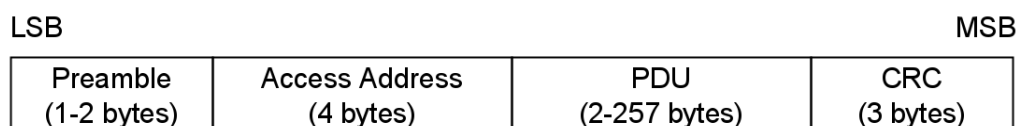
Stavový automat na linkové vrstvě může mít několik instancí. Díky tomu může být zařízení ve stejnou chvíli master i slave. Jak zařízení v roli master, tak i v roli slave mohou mít ve stejnou chvíli více navázaných spojení. Nutno dodat, že standard vyžaduje na linkové vrstvě alespoň jeden ze stavů advertising a scanning. Tím pádem se můžeme setkat se zařízeními, které umožňují ve stavu connection plnit pouze jednu roli. Význam výše uvedených stavů bude více objasněn v kapitole popisující GAP.

Identifikační adresa zařízení může být buď veřejná nebo náhodná. Má pevnou délku, a to 48 bitů. Veřejná adresa je neměnná a musí být registrována u organizace IEEE. Náhodná adresa může být předprogramována nebo náhodně generována za

běhu zařízení. Náhodná adresu může být statická, soukromá rozpoznatelná nebo soukromá nerozpoznatelná adresa. Výhodou náhodné adresy je, že nemusí být registrována u IEEE.

Formát paketu

Standard Bluetooth 5.0 definuje dva formáty paketu. Paket pro přenosové režimy LE 1M a LE 2M můžeme vidět na obr. 1.5. Paket pro režim dlouhého dosahu LE Coded je obdobný. Jeho strukturou se zde však zabývat nebudeme.



Obr. 1.5: Formát paketu Bluetooth 5.0

Paket pro režimy LE 1M a LE 2M se skládá ze 4 částí. Jeho formát je stejný jak při vysílání propagačních zpráv, tak i při výměně dat ve stavu connection. Jednotlivé části mají následující význam:

- Preamble (preamble) – Velikost záleží na přenosovém režimu (8 bitů při LE 1M, 16 bitů při LE 2M). Přijímač ji používá pro synchronizaci frekvence a času.
- Přístupová adresa (access address) – Slouží jako korelační kód přenášených dat. Pevnou adresu 0X8E89BED6 mají propagační zprávy. Pro všechny ostatní zprávy je adresa náhodně generována, přičemž po čas jednoho spojení mezi dvěma zařízeními se tato adresa nemění.
- PDU – Část paketu obsahující data. Struktura PDU se liší podle toho, jestli přenášíme propagační zprávy (advertising channel PDU) nebo datové zprávy (data channel PDU).
- Cyklický redundantní součet (CRC) – Kontrolní součet vypočítaný z PDU. Slouží pro odhalení chyb při přenosu.

1.5.3 Režim pro přímé testování (Direct Test Mode)

Slouží pro testování rádiového vysílání. Používá se při výrobě a certifikačních testech. V této práci se touto částí zabývat nebudeme.

1.5.4 Rozhraní hostitel-ovladač (HCI)

Protokol HCI zajišťuje komunikaci mezi částmi hostitel a ovladač. Ty mohou být fyzicky ve stejném čipu nebo mohou být ve fyzicky oddělených čipech. Pokud se na-

cháží ve stejném čipu, je protokol implementován přímo na čipu. Ve druhém případě je protokol implementován skrz jedno z rozhraní UART, USB nebo SDIO (Secure Digital Input Output).

1.5.5 Generic Access Profile (GAP)

Z hostitelské části architektury BLE se budeme nejdříve zabývat vrstvou Generic Access Profile. Vrstva definuje vysílání, skenování a propojení BLE zařízení. Jinými slovy GAP definuje, jak spolu BLE zařízení komunikují.

Vrstva GAP obsahuje 4 role, které může zařízení zastávat:

- Vysílač (broadcaster) – Je jím takové zařízení, které pouze vysílá propagační zprávy. Nepřijímá žádné data a neumožňuje navázat spojení s jiným zařízením. Vysílač odpovídá stavu advertising na linkové vrstvě. Výhodou je, že může více zařízení přijímat data ve stejnou chvíli, aniž by muselo dojít k navázání komunikace. Může jít např. o teploměr, který vysílá naměřenou teplotu pomocí propagačních zpráv.
- Pozorovatel (observer) – Pozorovatel může pouze přijímat propagační zprávy. Taktéž neumožňuje navázat spojení s dalším zařízením. Na linkové vrstvě odpovídá pozorovatel stavu scanning. Jako příklad můžeme uvést displej zobrazující teplotu, která byla přijata v propagační zprávě.
- Centrální (central) – Zařízení s touto rolí může přijímat propagační zprávy a data. Dále může být iniciátorem pro navázání komunikace, a tudíž umožňuje navázat spojení. Centrální role odpovídá roli master na linkové vrstvě.
- Periferní (peripheral) – Periferní zařízení může vysílat propagační zprávy a přijímat požadavky na navázání komunikace. Umožňuje navázat komunikaci. Periferní role na GAP vrstvě odpovídání roli slave na linkové vrstvě.

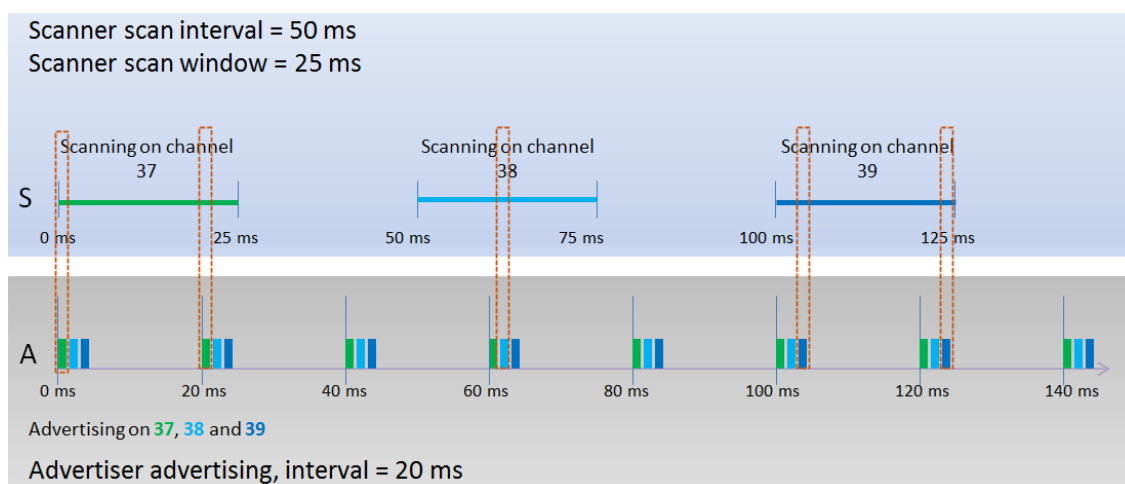
Jedno zařízení může zastávat ve stejnou chvíli více rolí. Mobilní telefon spojený s chytrými hodinkami zastává centrální roli. Pokud je telefon zároveň spojený s PC, tak zastává periferní roli vzhledem k PC. Role na GAP vrstvě se tedy vztahují k jednomu konkrétnímu spojení, nikoliv k jednomu zařízením.

Vysílání propagačních zpráv (advertising)

S vrstvou GAP souvisí stavy, které jsou popsány na linkové vrstvě. Nejdříve se budeme zabývat stavem advertising, ve kterém zařízení vysílá propagační zprávy. Vysílání probíhá na kanálech 37, 38, 39. Tyto tři kanály se také někdy nazývají jako primární kanály. Ve skutečnosti může vysílání propagačních zpráv probíhat i na všech ostatních kanálech, které bývají označovány jako kanály sekundární. Tato funkcionality byla implementována s příchodem Bluetooth 5.0. Vysílání na primárních kanálech umožňuje přenášet 31 bytů v jedné propagační zprávě, kdežto na

kanálech sekundárních můžeme přenést až 255 bytů. Budeme se zabývat pouze vysíláním na primárních kanálech.

Vysílání propagační zprávy můžeme chápat jako událost, která je vykonávána jednou za určitou dobu. Tato doba se skládá z tzv. propagačního intervalu (advertising interval) a ze zpoždění (advertising delay). Propagační interval může být v rozmezí 20 ms až 10,4 sec. Zpoždění v rozmezí 0 - 10 ms je náhodně generované. Při každé události se zpráva odvysílá na všech primárních kanálech.



Obr. 1.6: Vysílání a skenování propagačních zpráv [6]

Součástí vysílání propagačních zpráv je i odezva (scan response) na skenovací požadavek (scan request), který může vyslat zařízení nacházející se ve stavu scanning na linkové vrstvě. Obsahem této odezvy můžou být další užitečné informace. To opět umožňuje přenést více dat bez nutnosti propojení zařízení.

Skenování propagačních zpráv (scanning)

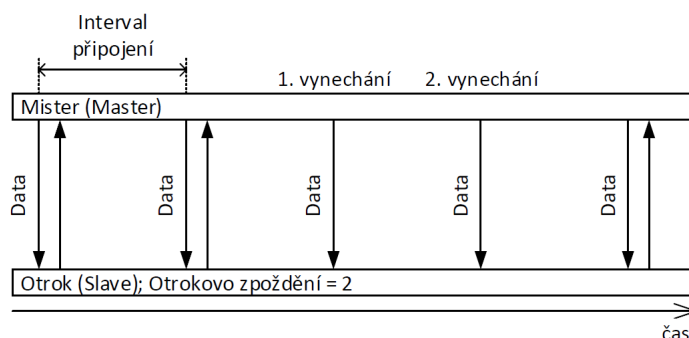
Zařízení se na linkové vrstvě nachází ve stavu scanning. Skenování probíhá na primárních kanálech a má podobné vlastnosti jako vysílání. Periodu, jak často dochází ke skenování, definuje skenovací interval (scan interval). Skenovací okno (scan window) udává, jak dlouho je konkrétní kanál skenován. Rozdíl oproti vysílání je ten, že počas události je skenován pouze jeden kanál. Zařízení může na propagační zprávu odpovědět buď požadavkem na spojení (connection request) nebo skenovacím požadavkem. Druhým zmíněným se většinou žádá o další informace od vysílacího zařízení. Pokud je umožněno vyslat skenovací požadavek, jedná se o aktivní skenování. V opačném případě jde o skenování pasivní.

Spojení zařízení

Z výše popsaného vyplývá, že ke spojení dvou zařízení je nutné, aby jedno z nich vysílalo propagační zprávy a druhé je skenovalo. Po přijetí propagační zprávy vysílá skenovací zařízení požadavek na spojení. Spojení je považováno za navázané až ve chvíli, kdy je od vysílacího zařízení přijat paket dat. V tuto chvíli přechází vysílací zařízení do role slave a skenovací do role master.

Parametry komunikace definuje master, který je může v průběhu komunikace změnit. O změnu může požádat i slave. Hlavními parametry komunikace jsou:

- Interval připojení (connection interval) – Definuje časový rozestup mezi dvěma komunikačními událostmi.
- Zpoždění slave (slave latency) – Počet po sobě jdoucích komunikačních událostí, které může slave ignorovat, aniž by byla komunikace považována za ukončenou.
- Časový limit dohledu (supervision timeout) – Jedná se o maximální čas mezi dvěma po sobě přijatými pakety. Pokud je tento čas překročen, je spojení považováno za ztracené.



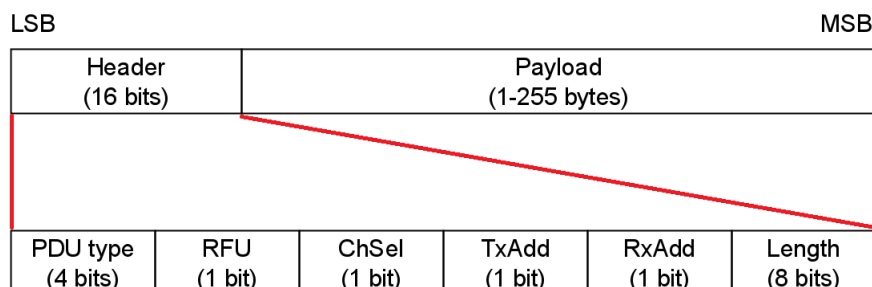
Obr. 1.7: Interval připojení a zpoždění slave [5]

PDU propagační zprávy

V kapitole Formát paketu bylo zmíněno, že část PDU se v paketu liší podle typu komunikovaných dat. Zaměříme se na situaci, kdy dochází k vysílání a skenování propagačních zpráv. PDU sestává z hlavičky o velikosti 16 bitů a z užitečné zprávy (Payload), která má proměnnou velikost. Parametry hlavičky jsou následující:

- Typ PDU (PDU Type) – Obsahuje informaci o typu PDU, který souvisí s rolí na GAP vrstvě. Dále záleží, jestli se vysílá na primárních nebo sekundárních kanálech.
- Rezervováno (RFU) – bit je rezervován pro budoucí použití

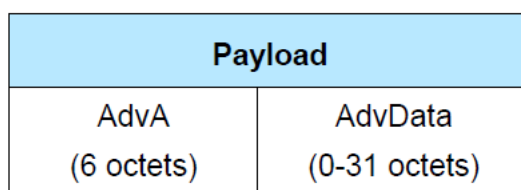
- ChSel, TxAdd, RxAdd – Jejich nastavení záleží na typu PDU a je popsáno ve specifikaci standardu. TxAdd a RxAdd značí, zda má konkrétní zařízení veřejnou nebo náhodnou adresu.
- Délka (Length) – Délka užitečné zprávy v bytech.



Obr. 1.8: PDU propagačních zpráv

Typ PDU v sobě nese informace o vlastnostech zařízení, které souvisejí s rolí na GAP vrstvě. Zařízení je skenovatelné (scannable), pokud může přijat skenovací požadavek. V opačném případě je neskenovatelné (nonscannable). Zařízení je připojitelné (connectable), pokud může přijat požadavek na připojení a tím pádem i navázat spojení. V opačném případě je nepřipojitelné. Pokud je propagační zpráva určena pro konkrétní zařízení, jedná se o přímou událost (directed). Pokud není adresát určen, jde o událost nepřímou (undirected).

Užitečná zpráva se skládá ze dvou částí. První část obsahuje adresu o velikosti 6 bytů (AdvA). Druhou část tvoří data nebo adresa. Ve většině případů je tvořena daty.

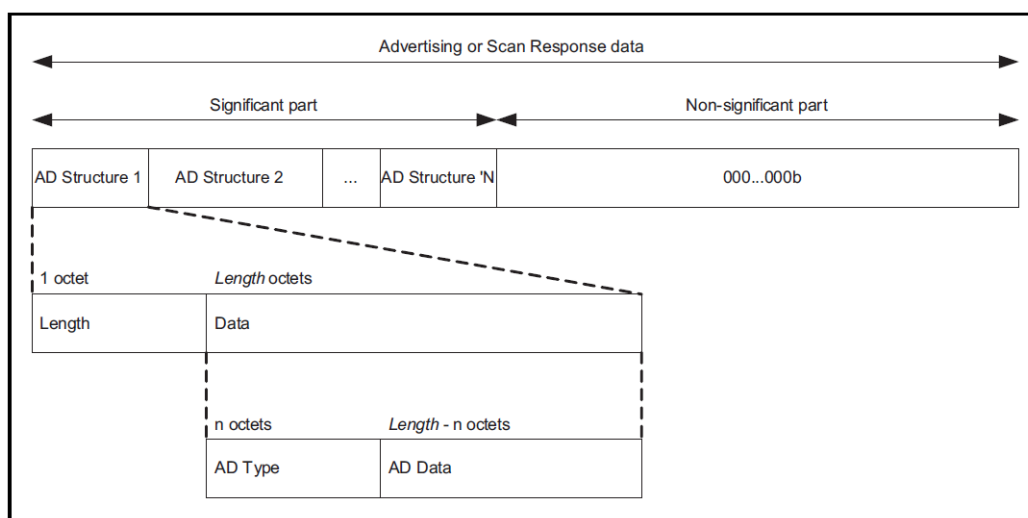


Obr. 1.9: Užitečná zpráva v PDU [2]

Pokud je druhá část tvořena daty, má následující parametry:

- Délka (Length) - Počet bytů části zprávy, která sama o sobě tvoří celek.
- Typ dat (AD Type) - Informace o typu dat. Výčet není součástí specifikace standardu, ale je možné jej dohledat na stránkách společnosti Bluetooth [7]. Jako příklad si můžeme uvést název zařízení (AD Type = 0x09) nebo nastavený vysílací výkon (0x0A).

- Data (AD Data) - Data s konkrétními hodnotami. Popis této datové části je v dodatečném souboru popisující standard Bluetooth. Po registraci do skupiny SIG je možné jej stáhnout z [25].



Obr. 1.10: Formát datové části v PDU [2]

1.5.6 Logical Link Control & Adaption Protocol (L2CAP)

L2CAP přebírá protokoly z vyšších vrstev. Ty následně fragmentuje tak, aby je bylo možné zapouzdřit do BLE paketu. L2CAP zajišťuje i opačnou funkci, kdy z nižší vrstvy přebírá pakety BLE, které následně rekonstruuje do formátů protokolů vyšších vrstev.

1.5.7 Attribute Protocol (ATT)

Na úrovni tohoto protokolu jsou definovány dvě role: server a klient. Protokol definuje, jak jsou mezi serverem a klientem vyměňovány data. Server vystavuje data, které obsahuje nebo řídí. Datům, které server vystavuje klientovi, se říká atributy. Atribut se skládá ze 4 částí:

- Popisovač (Attribute Handle) – Jedná se o 16 bitovou adresu, kterou server přiřadí atributu. V rámci jednoho spojení je jedinečná.
- Typ (Attribute Type) – Specifikuje, jaká informace je v atributu přenášena. Typ je označován zkratkou UUID (Universally Unique Identifier). Typy definované organizací Bluetooth SIG jsou 16 bitové a jejich seznam je volně k dispozici [8]. Uživatelem definovaný typ má velikost 128 bitů. Výhodou 16 bitových UUID je jejich kompatibilita skrz všemi platformami a také snížení velikosti vysílaného paketu [1].

- Hodnota (Attribute Value) – Hodnota vztahující se k danému typu.
- Oprávnění (Attribute Permissions) – Udává, zda lze z atributu číst, zapisovat do něj, zapnout u něj notifikaci nebo indikace. Oprávnění není zjistitelné skrz ATT, ale na GATT vrstvě.

Attribute Handle (2 bytes)	Attribute Type (2 or 16 bytes)	Attribute Value (variable length)	Attribute Permissions
-------------------------------	-----------------------------------	--------------------------------------	-----------------------

Obr. 1.11: Formát atributu

V rámci atributu lze provést tyto operace:

- Příkazy (Commands) – Vysílá je klient na server. Nevyžadují odpověď.
- Požadavky (Requests) – Vysílá je klient na server. Vyžadují odpověď.
- Odpovědi (Responses) – Po vyřízení požadavku je serverem zaslána ke klientovi.
- Oznámení (Notifications) – Server posílá oznámení klientovi, aby oznámil změnu konkrétní charakteristické hodnoty. Není vyžadováno potvrzení od klienta. Výhodou je, že klient nemusí vyčkávat a kontrolovat, jestli už se daná hodnota změnila. Principiálně se jedná o přerušení, se kterým se můžeme setkat např. u mikrokontrolérů.
- Indikace (Indications) – Podobná operace jako oznámení, jen je od klienta vyžadováno potvrzení.
- Potvrzení (Confirmations) – Posílá klient na server po úspěšném přijetí indikace.

1.5.8 Security Manager

Jak již název vypovídá, vrsta se stará o bezpečnost BLE. Zabezpečení komunikace není povinné. Hlavní funkce Security Manager jsou následující:

- Generování a ukládání sdílených tajných klíčů mezi zařízeními.
- Filtrování zařízení při vysílání propagačních zpráv, skenování a navazování spojení. Zařízení jsou filtrována pomocí tzv. whitelistů.
- Šifrování vyměňovaných dat.
- Ověření, že dvě zařízení sdílejí stejný tajný klíč.

1.5.9 Generic Attribute Profile (GATT)

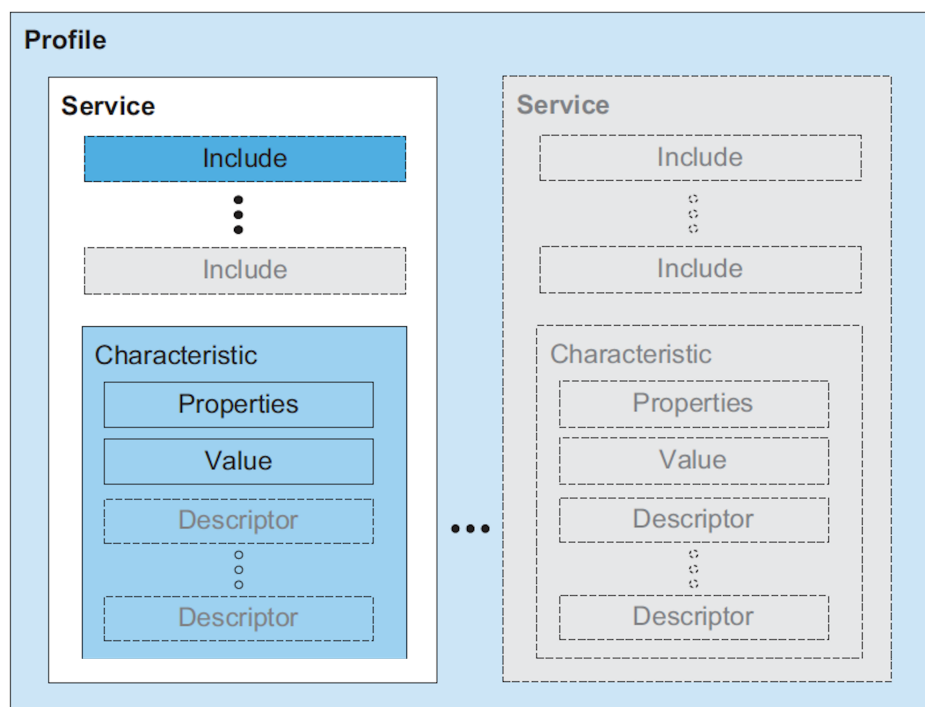
GATT definuje podobu komunikace po navázání spojení. Role server a klient mají stejný význam jako u ATT. Zařízení může svou roli měnit v závislosti na směru aktuální komunikace. Pro výměnu dat používá ATT, přičemž data jsou strukturována

do tzv. profilů (profile), služeb (services) a charakteristik (characteristics). Tyto tři jsou jednotlivé typy atributu, který byl popsán výše.

Charakteristiky obsahují vlastnosti (Properties), hodnotu (Value) a popisovače (Descriptor). Vlastnosti určují, jak může být hodnota použita (čtení, zápis, notifikace atd.). Popisovače můžou obsahovat popis charakteristiky, jednotky přenášené hodnoty a různé další. Hodnota se vztahuje ke konkrétní charakteristice. Můžeme tedy říct, že charakteristiky obsahují koncová uživatelská data. Jako příklad si uveďme charakteristiku *Temperature Measurement*, jejíž UUID=0x2A1C. Kromě měřené teploty může obsahovat informaci o jednotce teploty (°C nebo °F).

Služby seskupují atributy, které spolu nějak souvisí. Služby lze přirovnat ke třídám v objektově orientovaném programování. Častým příkladem je zapouzdření více charakteristik do služby.

Profily oproti službám navíc definují vztah mezi serverem a klientem. Obsahují seznam nutných služeb a charakteristik, podmínky navázání komunikace a parametry týkající se vysílání propagačních zpráv a spojení.



Obr. 1.12: GATT: Profily, služby, charakteristiky[2]

2 Bluetooth mesh

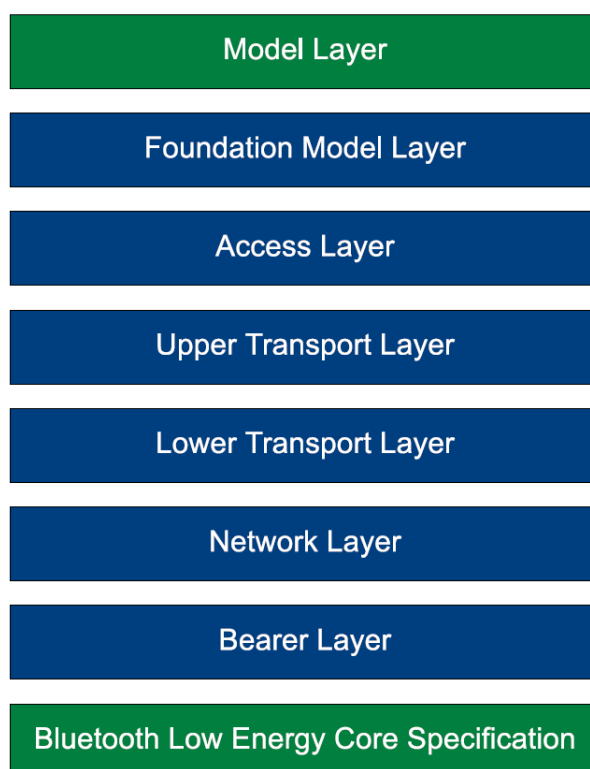
Před vydáním standardu pro Bluetooth mesh podporovalo BLE pouze dva typy síťové topologie. První z nich je one-to-one, kdy jsou spojeny dvě zařízení. Druhá je one-to-many, kdy se jedno zařízení chová jako broadcaster a zbylá zařízení přijímají data. S nástupem Bluetooth 5.0 přišla podpora síťové topologie many-to-many, kterou implementuje standard Bluetooth mesh [19].

Zařízení v síti mesh mezi sebou nenavazují komunikaci tak jako je tomu u zařízení BLE, s výjimkou tzv. proxy nodes. Zařízení se nacházejí ve stavech advertising a scanning, tudíž si zprávy vyměňují pomocí propagačních zpráv.

Velkou výhodou mesh je relativně velký dosah a odolnost sítě při výpadku některého ze zařízení. Bluetooth mesh je kompatibilní se všemi verzemi BLE, tedy s Bluetooth 4.0 a vyššími.

2.1 Architektura Bluetooth mesh

Na obr.2.1 je znázorněna architektura Bluetooth mesh, která je postavena nad architekturou BLE. Na Bluetooth mesh se však můžeme dívat i tak, že stojí po boku BLE, jelikož plně využívá pouze linkovou a fyzickou vrstvu BLE.



Obr. 2.1: Architektura Bluetooth mesh[19]

- **Nosná vrstva (Bearer layer)** – Definuje, jak jsou mezi jednotlivými uzly přenášeny zprávy. Jsou dva způsoby: *advertising bearer* a *GATT bearer*. Skrz *advertising bearer* komunikují uzly, které podporují mesh. Prostřednictvím *GATT bearer* je možné se sítí mesh spojit takové zařízení, které mesh nepodporují.
- **Síťová vrstva (Network layer)** – Zajišťuje adresování přenášených zpráv. Má za úkol filtrovat zprávy, které nejsou pro konkrétní uzel určeny, případně tyto zprávy přeposílat dále. V neposlední řadě se stará o šifrování na síťové vrstvě.
- **Transportní vrstva (Transport layer)** – Segmentování objemných zpráv a jejich zpětné seskupení zajišťuje transportní vrstva. Na této vrstvě je dále řešeno šifrování a ověřování aplikačních dat, která jsou vyměňována s přístupovou vrstvou.
- **Přístupová vrstva (Access layer)** – Definuje formát aplikačních dat. Na základě aplikačního klíče přeposílá data do vyšších vrstev.
- **Modelová vrstva (Model layer)** – Vrstva definuje zprávy, stavy a modely, které jsou v mesh síti používány.

2.2 Prvky a vlastnosti Bluetooth mesh

Součástí standardu Bluetooth mesh jsou nové prvky a vlastnosti, se kterými se u BLE nesetkáme. Ty si nyní popíšeme a vysvětlíme jejich funkcionalitu. Pro některé z nich bude ve zbytku práce používán jejich anglický termín.

2.2.1 Uzly, elementy, modely

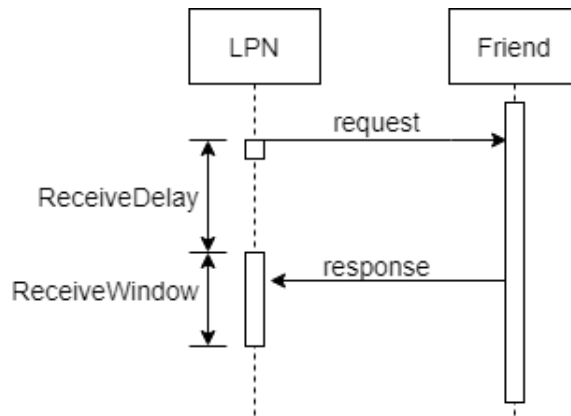
Uzly (nodes)

Zařízení, které jsou součástí sítě mesh, se říká uzly. Tzv. neprovozované zařízení (*unprovisioned device*) je takové zařízení, které není součástí mesh. Přidání zařízení do mesh se říká *provisioning* a bude popsáno později. Jedna síť mesh může obsahovat až 32 767 uzlů. Každý uzel může odesílat a přijímat data. Podle svých schopností se dále dělí na:

- **Relay nodes** – Uzly, které přeposílají přijaté zprávy. Maximální počet přeposlání (nebo také přeskoků) jedné konkrétní propagační zprávy definuje TTL (*Time To Live*). Jedná se o celočíselnou hodnotu. Je podporováno maximálně 127 přeskoků.
- **Proxy nodes** – Jsou to uzly, které poskytují rozhraní mezi Bluetooth mesh a BLE prostřednictvím vrstvy GATT. Pomocí proxy nodes tak můžeme nepřímo

propojit se sítí mesh zařízení, které nepodporují mesh [9]. Pro komunikaci jsou použity GATT služby *Mesh Provisioning* (UUID=0x1827) a *Mesh Proxy* (UUID=0x1828).

- Friend nodes a low power nodes (LPN) – Tyto dva typy uzlů naleznou své uplatnění zejména v aplikacích, kde cílíme na nízkou spotřebu energie. Navázání spolupráce mezi friend a LPN je ve specifikaci označováno *friendship*. Friend shromažďuje zprávy, které jsou určeny pro LPN, se kterým navázal spolupráci. LPN je většinu času v režimu spánku. Data vysílá a přijímá pouze ve stanovených intervalech, které jsou vyjednány při navazování spolupráce. Výměnu dat si popíšeme na obrázku níže. LPN po probuzení pošle žádost případně další data, a vrátí se zpět do spánku. Parametr *ReceiveDelay* definuje dobu, za jak dlouho začne od odeslání požadavku přijímat data. Doba, jak dlouho je LPN aktivní a tedy přijímá data, definuje parametr *ReceiveWindow*. V průběhu této doby musí friend vystavit data pro LPN. Po přijetí dat může LPN výměnu opět zahájit nebo přejít do spánku.



Obr. 2.2: Spolupráce mezi LPN a friend node

Elementy

Elementy tvoří uzel. Každý uzel má minimálně 1 element. Jako příklad si uvedeme inteligentní stropní svítidlo s více LED diodami. Každá LED dioda představuje 1 element, zatímco stropní svítidlo je chápáno jako 1 uzel.

Modely

Model je část, která zajišťuje jednu nebo více funkcionalit elementu. Každý element musí obsahovat alespoň jeden model. Modely jsou definovány buď skupinou Bluetooth SIG, přičemž identifikátor má 16 bitů, nebo si je může uživatel definovat sám. Ve druhém případě bude mít identifikátor 32 bitů. Jako příklad si uvedeme

jednoduchý model *Generic ON/OFF Server*, který může vypínat a zapínat LED diodu, resp. jakýkoliv dvoustavový výstup. V IoT často uplatníme modely pro senzory (např. *Sensor Server model*), které jsou univerzální a nezáleží tak na typu a vlastnostech senzorů.

2.2.2 Stav

Každý model obsahuje několik stavů. Význam termínu *stav* ve spojení s Bluetooth mesh však může být pro čtenáře matoucí. Pro představu zmíníme stavy pro *Sensor Server model*. Prvním z nich je *Sensor Descriptor* a obsahuje informace o měřené veličině. Typ a formát dat je uložen v *Sensor Property ID*. Hexadecimální kód pro různé veličiny můžeme najít ve specifikaci [21]. *Sensor Negative Tolerance* a *Sensor Positive Tolerance* definují minimální a maximální hodnoty měřené veličiny. *Sensor Server model* obsahuje ještě další parametry, které tento stav popisují. Druhý stav pro *Sensor Server model* je *Sensor Data*. Obsahuje měřená data v předem stanoveném formátu [25].

Stav tedy popisuje jednu nebo více konkrétních vlastností a hodnot, které se vztahují k modelu. Stavy můžou být určeny buď jen ke čtení nebo ke čtení a zápisu.

2.2.3 Zprávy

Komunikace v rámci mesh probíhá pomocí zpráv (messages). Zprávy se dělí na řídicí (control messages) a přístupové (access messages). Řídicí zprávy se týkají provozu sítě mesh. Může jít např. o požadavek navázání spolupráce mezi friend node a LPN. Přístupové zprávy dále rozlišujeme podle operačního kódu (opcode), který značí některou z následujících operací:

- Get – Slouží pro vyžádání stavu uzlu v síti.
- Set – Slouží pro změnu hodnoty daného stavu.
- Status – Zpráva se používá ve více případech. Prvním z nich je odezva na Get zprávu. Další je potvrzení Set zprávy. Poslední případ je poslání zprávy, aniž by byly obslužovány Get/Set zprávy. Toto může být vyvoláváno např. periodicky.

2.2.4 Adresy

Součástí přidání zařízení do mesh sítě je i přidělení mesh adres pro toto zařízení. Rozlišujeme tři typy adres: unicast, skupina (group) a virtuální (virtual). Zatímco mesh adres může mít uzel několik, Bluetooth MAC adresu má pouze jednu.

Unicast adresa je jedinečná pro každý element. Každý uzel má přidělený určitý rozsah těchto adres, aby pokryl implementované elementy. V jedné síti může být maximálně 32 767 unicast adres.

Skupinová adresa se používá pro adresování určité skupiny uzlů. Skupina často odpovídá fyzickému seskupení, např. všechna světla v patře rodinného domu. Při odchodu z domu pak můžeme najednou zhasnout všechna světla. Také standard definuje některé skupinové adresy, přičemž skupinu tvoří konkrétní typ uzlů (proxy, relay atd.).

Virtuální adresy jsou podobné skupinovým adresám. Výrobci je používají pro přednastavení, aby bylo možné jejich produkt adresovat globálně [10].

2.2.5 Komunikace v rámci mesh

Model publish-subscribe

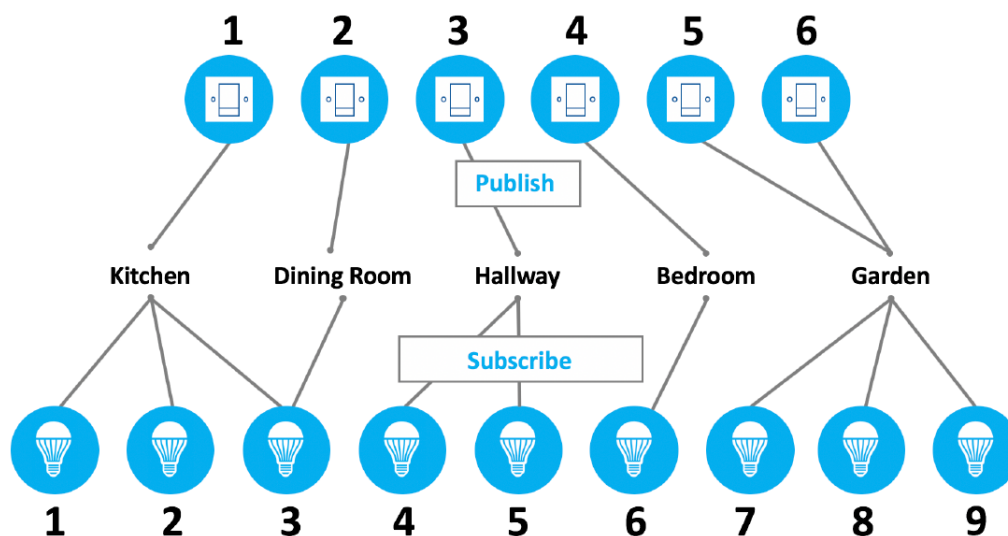
Pro výměnu zpráv v rámci sítě mesh je použit tzv. publish-subscribe model. Ten spočívá v publikování (publish) a odebírání (subscribe) zpráv. Posílání zpráv bývá v anglické literatuře označováno publishing. Aby mohlo zařízení přijímat zprávy z konkrétních adres, je třeba jej vhodně nakonfigurovat. Této konfiguraci se říká subscribing. Na obrázku níže je ukázka modelu publish-subscribe. Světlo č. 3 odebírá zprávy ze skupin *Kitchen* a *Dining Room*. Vypínače 5 a 6 publikují zprávy pro skupinu *Garden*, která obsahuje světla 7-9.

Tab. 2.1: Skupiny pro jednotlivé typy uzlů

Adresa skupiny	Adresované uzly
0xFFFC	proxy nodes
0xFFFD	friend nodes
0xFFFE	relay nodes
0xFFFF	all nodes

Řízení toku zpráv

Již bylo zmíněno, že zpráva obsahuje informaci o maximálním počtu přeskoků (TTL). Uzel, který obdrží zprávu, která pro něj není určena, přepośle tuto zprávu všem dalším uzlům v jeho dosahu. Každým přeposláním zprávy se TTL dekrementuje. Ve chvíli, kdy se rovná 0, zpráva zaniká. Uzel ignoruje zprávu, kterou už jednou obdržel. Zprávy mohou přeposílat pouze uzly relay. Pro robustnost sítě je tedy nutné vhodné rozmístění a dostatečný počet těchto uzlů.



Obr. 2.3: Ukázka modelu publish-subscribe [1]

2.3 Zabezpečení v Bluetooth mesh

Zabezpečení Bluetooth mesh je povinné. Zabezpečení sítě, aplikace a zařízení je provedeno samostatně a používá tyto klíče:

- **Síťový klíč (NetKey)** – Klíč je stejný pro všechny uzly, které jsou součástí jedné sítě mesh. Slouží pro dešifrování a ověřování zpráv na síťové úrovni.
- **Aplikační klíč (AppKey)** – Vztahuje se ke konkrétní aplikaci v síti a může jej sdílet více uzlů. Jako příklad můžeme uvést mesh síť vybudovanou v hotel: 1. aplikační klíč bude náležet zařízením, které používá personál, 2. aplikační klíč pak bude přidělen zařízením, které jsou určeny pro zákazníky hotelu. Tím zajistíme rozdílné oprávnění pro personál a zákazníky.
- **Klíč zařízení (DevKey)** – Klíč používaný pro zabezpečení komunikace mezi provisionerem a uzlem. Pouze provisioner má tento klíč k dispozici [26].

Další z prvků zabezpečení je ochrana proti opakovanému posílání zpráv. Každý paket obsahuje *sequence number* a *IV index*. Sequence number je při každém odeslání zprávy inkrementováno. Dohromady s IV indexem dávají unikátní číslo (*nonce*). Pokud zpráva obsahuje hodnoty sequence number a IV indexu menší nebo rovny předešlé zprávě, je zpráva smazána.

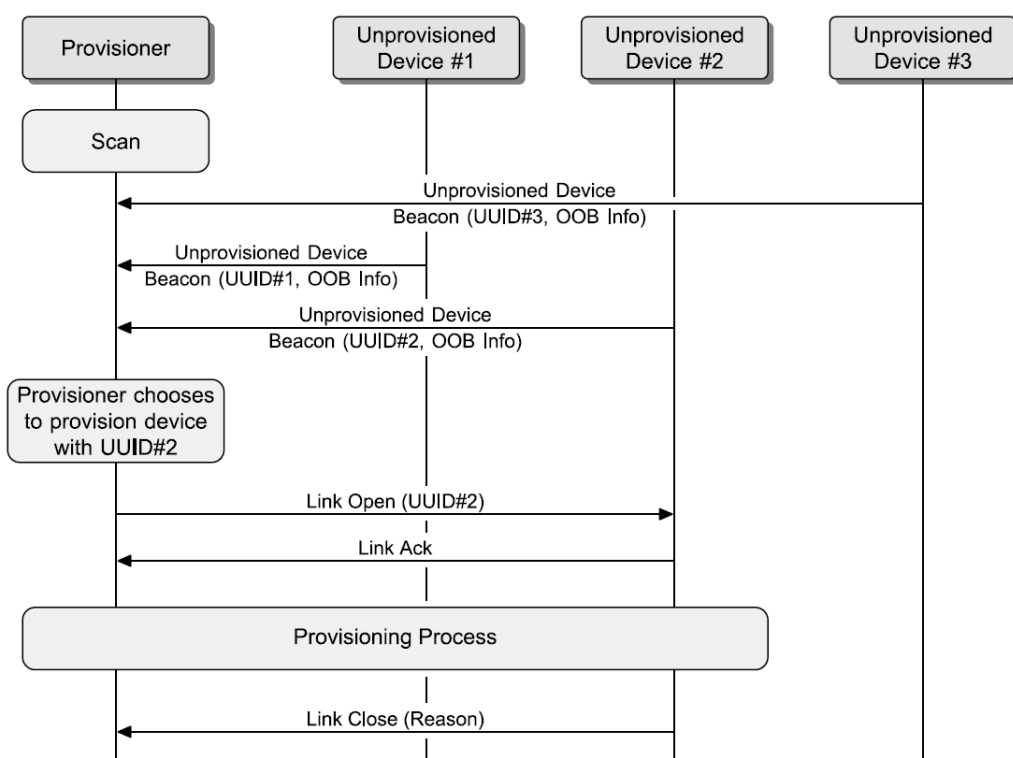
2.4 Přidání zařízení do mesh

Zařízení může být přidáno jiným uzlem v síti nebo zařízením propojeným skrz proxy node (např. mobilní telefon). Zařízení, které přidává do sítě nové uzly, se nazývá *provisioner*. Zařízení, které není součástí žádné sítě mesh, se nazývá neprovozované za-

řízení (unprovisioned device). Přidání zařízení je ve specifikaci označováno termínem *provisioning* a lze jej shrnout do následujících bodů:

- Neprovozované zařízení vysílá zprávy o své přítomnosti, které obsahují UUID zařízení. Zprávy vysílá buď přes *advertising bearer* nebo přes *GATT bearer*.
- Provisioner posílá pozvání do mesh sítě, na které neprovozované zařízení odpovídá zprávou obsahující počet elementů zařízení, podporované zabezpečovací algoritmy a metody ověření.
- Následně dochází k výměně veřejných klíčů a ověření neprovozovaného zařízení. Jedna z možných metod ověření je tzv. *Output OOB (Out of Band)*. Neprovozované zařízení podle svých možností signalizuje náhodně vygenerované číslo (blikání LED diody, zobrazení na displeji atd.). Uživatel spravující provisionera toto číslo zadá, přičemž dojde k ověření shody.
- Po ověření posílá provisioner další data, která obsahují síťový klíč, klíč zařízení, unicast adresu přidělenou prvnímu elementu a IV index. V tomto okamžiku se zařízení stává provozovaným zařízením a zároveň uzlem sítě mesh.

Zařízení musí být po přidání do sítě nakonfigurováno, jinak nebude vykazovat požadovanou funkčnost. Konfiguraci zajišťuje provisioner. Mezi provisionerem a cíleným uzlem jsou vyměněny informace o dostupných modelech. Provisioner nastaví publish a subscribe adresy modelů, periody publikování, TTL a další.



Obr. 2.4: Přidání neprovozovaného zařízení do sítě mesh [19]

3 Měření oxidu uhličitého

Měření koncentrace oxidu uhličitého v ovzduší se provádí z několika různých důvodů. Cílem této práce je monitorování CO_2 za účelem zajištění podmínek hygieny. Limitní hodnoty jsou stanoveny nařízením vlády [11]. Koncentrace se nejčastěji uvádí v jednotkách ppm (z anglického parts per million).

Tab. 3.1: Vliv CO_2 na lidský organismus [12]

Koncentrace CO_2	Komentář, příznaky
< 400 ppm	koncentrace ve venkovním vzduchu
< 1000 ppm	doporučená úroveň CO_2 ve vnitřním prostředí
< 1500 ppm	maximální doporučená úroveň CO_2 ve vnitřním prostředí
> 1500 ppm	příznaky únavy, snižování koncentrace, ospalost, letargie
< 5000 ppm	maximální bezpečná koncentrace CO_2 bez zdravotních rizik
> 5000 ppm	příznaky nevolnosti, bolesti hlavy, zvýšený tep
> 10 000 ppm	při dlouhodobém působení prokázány zdravotní problémy
> 40 000 ppm	životu a zdraví nebezpečná koncentrace

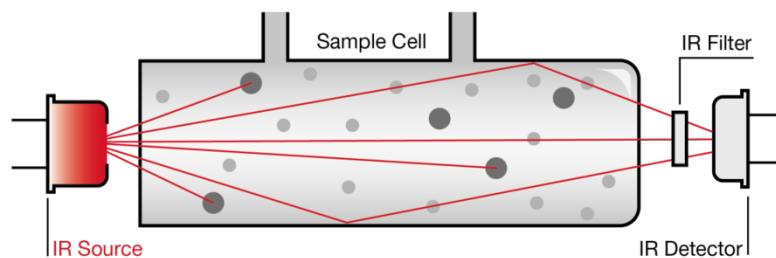
3.1 Snímače oxidu uhličitého

Pro měření koncentrace CO_2 je k dispozici několik principů. Nejrozšířenější jsou nedisperzivní infračervené snímače (NDIR). Dále se můžeme setkat se snímači, které pro vyhodnocení používají elektroakustický nebo elektrochemický princip.

Snímače NDIR

NDIR (NonDispersive Infrared) snímače obsahují zdroj infračerveného záření, světlo-vodnou trubici a detektor infračerveného záření s filtrem. Měřený plyn prochází světlo-vodnou trubicí. IR záření je absorbováno molekulami CO_2 . Díky tomu dochází k útlumu vysílaného záření. Signál z detektoru záření je zesílen a poté vyhodnocen útlum, který udává koncentraci měřené látky, v našem případě CO_2 .

Snímače NDIR se vyznačují selektivitou, dlouhou stabilitou a životností. Díky těmto vlastnostem je jejich vyšší pořizovací cena. Neýhodou může být rozsah, kdy více molekul sledované látky může absorbovat prakticky veškeré záření. Přesnost pod $\pm 5\%$ můžeme očekávat u senzorů s měřicími rozsahy v řádech tisíců ppm [12].



Obr. 3.1: NDIR snímač CO₂ [13]

Elektroakustické snímače

Elektroakustické snímače vyhodnocují změnu kmitočtu ultrazvuku v mechanickém rezonátoru. Na základě změn kmitočtu ultrazvukových vln je elektronikou vyhodnocena koncentrace CO₂ ve vzduchu. Výhodou snímačů je dlouhodobá stabilita bez nutnosti překalibrování.

Elektrochemické snímače

Elektrochemický snímač se skládá z článku s elektrolytem, elektrod a elektroniky. Materiál elektrod, napětí mezi elektrodami a složení elektrolytu jsou zvoleny tak, aby na měřicí elektrodě docházelo k elektrochemické reakci, při níž vznikají volné elektrony. Zesílený elektrický proud odpovídá koncentraci CO₂.

Tyto snímače jsou poměrně přesné, bohužel mají krátkou životnost, přibližně 1 až 2 roky. Z důvodu elektrochemických změn v elektrolytu vyžadují častou kalibraci.

3.2 Vybrané snímače oxidu uhličitého

Pro měření koncentrace CO₂ byly vybrány dva moduly z podobné cenové kategorie, a to MH-Z14A a MH-Z19B. Oba využívají NDIR princip. Snímače mají v podstatě shodné vlastnosti. Při spuštění vyžadují predehřev, který může mít oproti normálnímu běhu větší proudový odběr. To může činit potíže zejména v low power aplikacích. Uživatel má možnost si snímače zkalibrovat. Parametry vybraných snímačů jsou uvedeny níže v tabulce.

Tab. 3.2: Parametry vybraných snímačů

Parametr	MH-Z14A [29]	MH-Z19B [28]
Pracovní napětí (DC)	4.5-5.5V	4.5-5.5V
Odběr proudu	< 60mA	< 20mA
Rozsah	0-10 000ppm	0-10 000ppm
Přesnost	podle zvoleného rozsahu	$\pm(50\text{ppm}+5\% \text{ z měřené hodnoty})$
Doba přehřevu	3min	3min
Výstupní signál	PWM, UART, 0.4-2V _{DC}	PWM, UART, 0.4-2V _{DC}
Doba odezvy	< 90s	< 60s
Pracovní teplota	0-50 °C	0-50 °C
Pracovní vlhkost	0-95%	0-95%
Životnost	5 let	5 let

4 Použité MCU

Celá řada výrobců nabízí mikrokontroléry resp. vývojové desky, které umožňují implementaci bezdrátových komunikací Bluetooth 5, ZigBee a Thread. V našem případě byly zvoleny vývojové desky od společností STMicroelectronics a Nordic Semiconductor. Mikrokontroléry, kterými jsou desky osazeny, disponují dvěma jádry postavené na architektuře ARM. První jádro Cortex-M0+ slouží pro implementaci některé z výše uvedené bezdrátové komunikace a dále zajišťuje rádiové vysílání 2.4 GHz. Druhé jádro Cortex-M4F slouží pro uživatelskou aplikaci a obsluhu periférií. Mikrokontroléry jsou vhodné pro low power aplikace.

4.1 STMicroelectronics

Od společnosti STMicroelectronics byla vybrána vývojová deska P-Nucleo-WB55 včetně její USB Dongle verze. Obě desky jsou vybaveny mikrokontrolérem z rodiny STM32WB, přičemž maximální frekvence jádra je 64 MHz. Nejvýkonnější čipy z této rodiny mají až 1 MB flash paměti a 256 KB SRAM paměti. Obsahují širokou škálu periférií.

P-Nucleo-WB55

Vývojová deska Nucleo68 je osazena mikrokontrolérem STM32WB55RG. Kromě uživatelských tlačítek, LED diod a velkého počtu vyvedených pinů obsahuje debugger ST-LINK, který slouží pro nahrávání kódu do paměti a krokování programu. Díky slotu na baterii CR2032 může deska posloužit jako plnohodnotná vývojová deska např. pro aplikace s BLE, případně Bluetooth mesh.

Varianta P-Nucleo-WB55 USB Dongle je osazena mikrokontrolérem STM32WB55CG. Má menší počet uživatelských tlačítek a méně vyvedených pinů. Zásadní rozdíl je však v absenci debuggeru. Nahrávání kódu je možné i bez přídavných programátorů. Varianta USB Dongle obsahuje kromě antény na plošném spoji také konektor UFL pro zapojení externí antény.

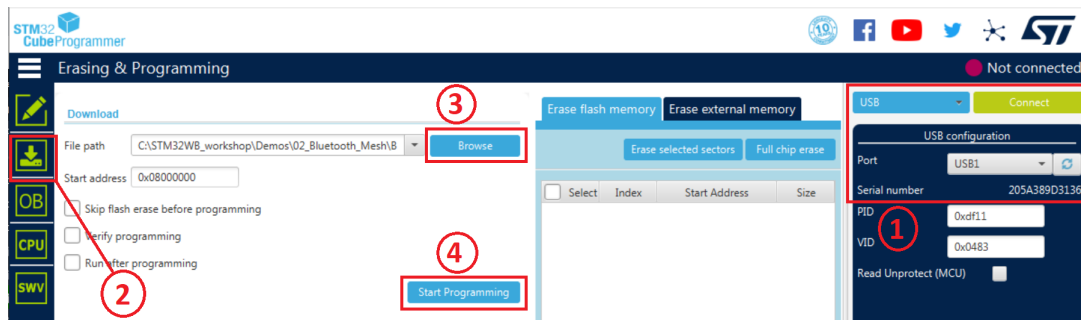
Vývojové prostředí

Pro vyvíjení aplikací je k dispozici několik vývojových prostředí. Z placených uvedeme IAR Embedded Workbench nebo Keil MDK. STMicroelectronics nabízí své open-source vývojové prostředí STM32CubeIDE, což je prostředí postavené na Eclipse.

Pokud nemáme k dispozici debugger a chceme programovat P-Nucleo-WB55 USB Dongle, vystačíme si s programem STM32CubeProgrammer, který je také volně

k dispozici. Před zasunutím desky do USB portu přepneme přepínač *SW2* do polohy 1, čímž přepneme mikrokontrolér do programovacího módu. Následující instrukce odpovídají krokům na obr. 4.1:

1. Výběr rozhraní USB, volba konkrétního portu, připojení se k mikrokontroléru
2. Aktivování záložky *Erasing, Programming*
3. Volba binárního souboru
4. Nahrání binárního souboru do paměti



Obr. 4.1: Programování P-Nucleo-WB55 USB Dongle

V neposlední řadě je také k dispozici softwarový balíček SDK (z anglického Software Development Kit) pro rodinu mikrokontrolérů STM32WB [15]. Obsahuje nízkourovňově ovladače (LL - low-layer), vrstvu hardwarové abstrakce (HAL - Hardware Abstraction Layer), sadu komponent pro middleware a různé další. V případě implementace některé z bezdrátových komunikací (Bluetooth, Zigbee, Thread) je vhodné zkontrolovat a případně aktualizovat binární soubor pro koprocessor. Postup a binární soubory nalezneme v balíčku SDK v adresáři `..\Projects\STM32WB_Copro_Wireless_Binaries\STM32WBxx`.

4.2 Nordic Semiconductor

Od Nordic Semiconductor byl vybrán mikrokontrolér nRF52840. Konkrétně se jedná o variantu USB Dongle, která neobsahuje debugger. Mikrokontrolér má prakticky totožné parametry jako výše zmíněné čipy z rodiny STM32WB. Lišit se můžou v některých perifériích, které pro nás nejsou podstatné.

Vývojové prostředí a programování

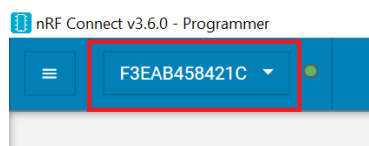
Opět máme možnost vybrat některý z placených programů (IAR, Keil) nebo použít volně dostupné vývojové prostředí SEGGER Embedded Studio. Po jeho stažení a bezplatném získání licence nejsme nijak omezeni co se týká velikosti kódu. Velmi

užitečný program je nRF Connect obsahující několik nástrojů pro testování a vývoj aplikací s BLE.

Zatímco SDK balíček od STM zahrnuje jak BLE, tak i Bluetooth mesh, Nordic Semiconductor vydává SDK balíček zvlášť pro BLE[24] a zvlášť pro Bluetooth mesh[23]. Abychom mohli používat SDK pro Bluetooth mesh, musíme nastavit cestu k BLE SDK. Tento postup platí pouze pro SEGGER Embedded Studio, jiné IDE nebylo použito. V nabídce *Tools* vybereme *Options*. Vyhledáme položku *Building - Global Macros* a doplníme makro `SDK_ROOT=cesta` tak, aby *cesta* byla adresář obsahující SDK BLE.

Při implementaci některého z bezdrátových protokolů musí být v paměti nahráný tzv. SoftDevice. Ten musí odpovídat zvolenému mikrokontroléru [17]. nRF52840 má SoftDevice S140 a náleží mu identifikační číslo 0x100, které bude později použito při nahrávání programu. Binární soubor SoftDevice s příponou `.hex` najdeme ve staženém balíčku SDK v adresáři `..\components\softdevice\s140`.

Kvůli absenci debuggeru na desce nRF52840 USB Dongle je třeba pro nahrání kódu použít tzv. Device Firmware Update (DFU) mód. Pokud je deska připojena do USB portu, stisknutím tlačítka *RESET* aktivujeme DFU mód. V programu nRF Connect otevřeme nástroj Programmer. V levém horním rohu vybereme COM port, který odpovídá připojené desce. V pravé části jsou panely pro výběr binárních souborů a programování/mazání paměti. Můžeme také vidět aktuální obsazení paměti a rozložení paměti již vybraných binárních souborů.



Obr. 4.2: Mód DFU - výběr COM portu

Již bylo zmíněno, že aplikace s bezdrátovými protokoly vyžadují SoftDevice. Pokud jej mikrokontrolér neobsahuje, musíme ho nahrát buď zvlášť nebo zároveň s aplikací. V případě, že je SoftDevice v paměti nahráný, tak jsme před samotným nahráním kódu vyzváni, abychom zadali identifikační číslo SoftDevice nahráného v paměti. Do pole *Custom SoftDevice ID* vyplníme 0x100 a potvrdíme.

Více informací k vývojovému prostředí, stažení SDK a programování mikrokontrolérů můžeme najít v podrobnějším popisu na stránkách Nordic Semiconductor [16].

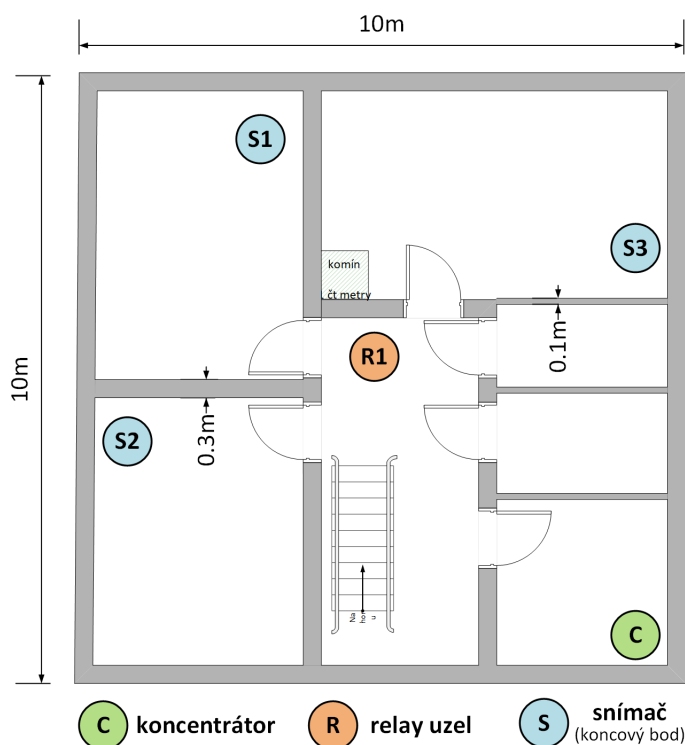
5 Návrh měřicího systému

V této kapitole budou nejdříve popsány podmínky, které bylo nutné vzít v potaz při návrhu měřicího systému. Čtenáře seznámíme s prostory, ve kterých byl bezdrátový systém realizován. Poté si rozebereme architekturu sítě z hlediska Bluetooth mesh a popíšeme dílčí prvky systému.

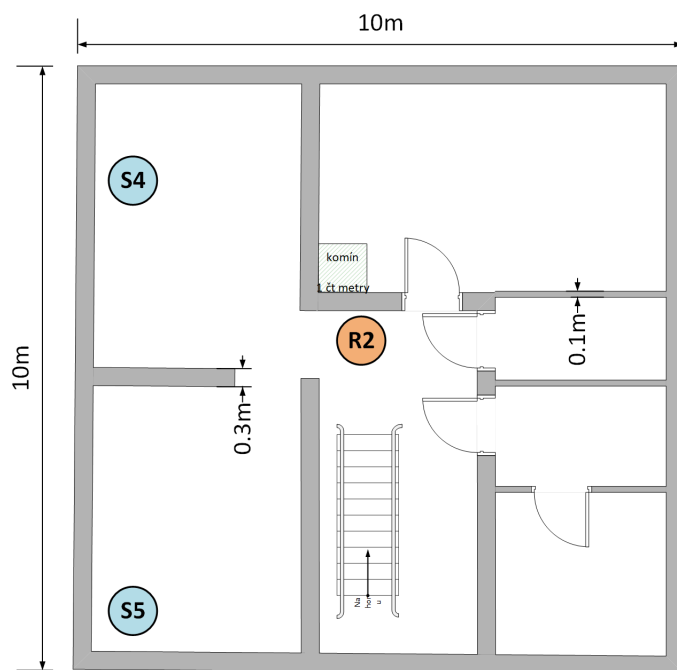
5.1 Podmínky a prostory pro měřicí systém

Před samotným návrhem měřicího systému byly vybrány mikrokontroléry resp. vývojové desky od společností STMicroelectronics a Nordic Semiconductor. Podrobnější rešerše nebyla předem provedena, což se posléze odrazilo při samotné realizaci měřicího systému. K dispozici bylo po dvou kusech STM32WB55RG (*Nucleo68*) a STM32WBWB55CG (*WB55 USB Dongle*), dále pak 4 kusy nRF52840 USB Dongle. Množství senzorů CO₂ bylo také velmi omezené: 1 kus MH-Z14A a 1 kus MH-Z19B.

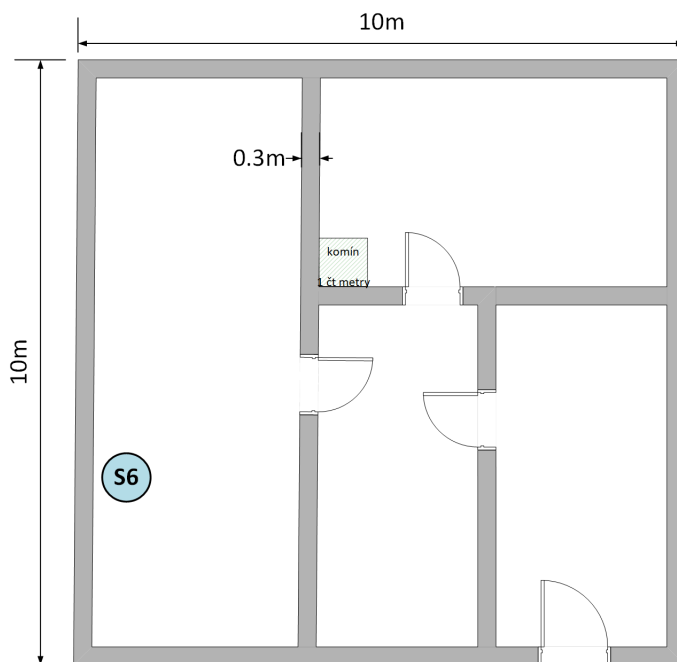
Práce byla realizována v domácích podmínkách. Na obrázcích 5.1 až 5.3 jsou orientační náčrty půdorysů jednotlivých podlaží domu. Všechny stěny jsou zděné a jejich šířka je 10 cm nebo 30 cm. Výška jednoho podlaží je 250 cm a výška železobetonového stropu je přibližně 30 cm.



Obr. 5.1: Půdorys 3.NP



Obr. 5.2: Půdorys 2.NP

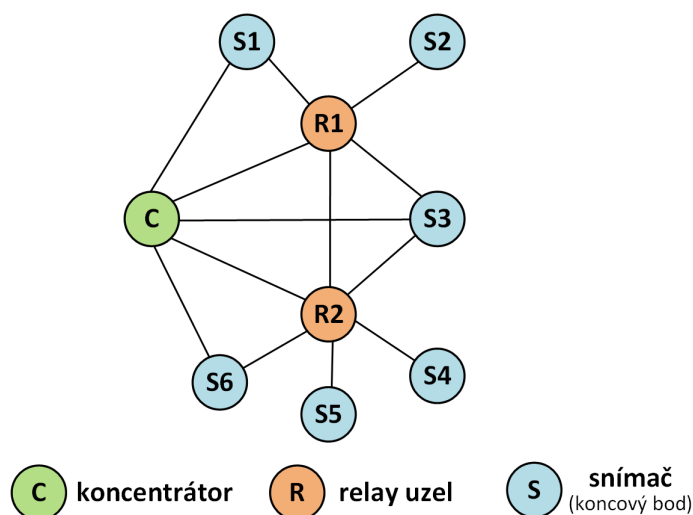


Obr. 5.3: Půdorys 1.NP

5.2 Architektura systému z pohledu Bluetooth mesh

Náčrty půdorysů také zahrnují předpokládané rozmístění jednotlivých prvků sítě mesh v domě. Jednotlivé koncové body jsou situovány do takových míst, kde se předpokládá zvýšená koncentrace CO₂ (kuchyň, pokoje atd.). Propojení systému z hlediska Bluetooth mesh znázorňuje obr.5.4.

Princip fungování celého systému je prostý. Koncové body měří koncentraci CO₂ a s použitím vhodného modelu odesílají měřená data do nadřazeného koncentrátoru. Měřicí zařízení nemusí být nutně koncový bod, podle potřeby mu může být programově předem přidána funkce relay uzlu. Ve vytvořené síti mesh se nenachází žádný LPN, a to kvůli tomu, že by v kombinaci s použitými senzory CO₂ postrádal svůj význam. Navzdory tomu se v další kapitole krátce zaměříme na některé změřené parametry LPN. LPN by bezesporu mohl být uplatněn v případě rozšíření sítě o další prvky.



Obr. 5.4: Mesh síť měřicího systému

5.2.1 Modely a konfigurace adres

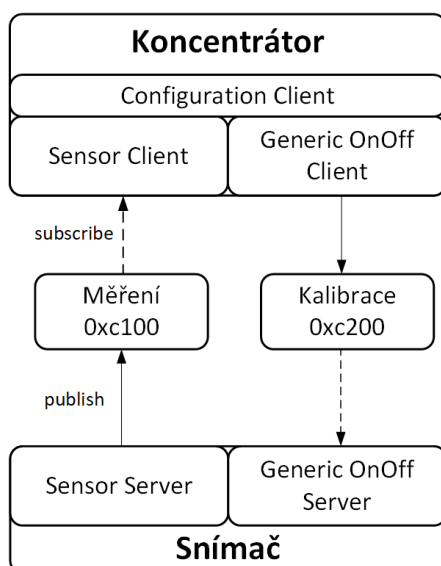
Jeden z požadavků byl, aby bylo možné mesh síť konfigurovat a spravovat pomocí tzv. *embedded provisionera*. To znamená, že síť není spravována aplikací v mobilním telefonu nebo tabletu, ale přímo z mikrokontroléru. Můžeme se setkat s rozdílnou terminologií pro tento typ provisionera. Embedded provisioner musí obsahovat *Config Client* model, jinak by nebylo možné provádět úkony spjaté se správou mesh sítě jako jsou např. přidávání/odebírání zařízení ze sítě, přidělování/odebírání adres pro publikování/odebírání zpráv a jiné. V případě Config Client modelu se nenastavují

žádné adresy pro publikování a odebírání zpráv. Koncentrátor našeho systému bude zastávat funkci embedded provisionera.

Pro posílání měřených hodnot použijeme *Sensor Server* model. Ten bude implementován na každém koncovém bodu a dále na uzlech relay, a to z důvodu většího vytížení sítě při měření parametrů sítě. Model bude publikovat do skupinové adresy 0xc100. Naměřené hodnoty se ukládají do stavu *Sensor Data*. Property ID pro měření koncentrace CO₂ je 0x0077 [21]. Hodnota je reprezentována charakteristikou *CO₂ Concentration*, která je datového typu uint16 [25].

Příjem dat na koncentrátoru zajistí *Sensor Client* model. Ten v našem případě nevyžaduje žádnou konfiguraci kromě přidělení adresy pro odběr zpráv, která je 0xc100.

Použité snímače můžou být kalibrovány uživatelem. Jeden ze způsobů je připojit pin *HD* na napětí 0V po dobu nejméně 7 sekund. Pro umožnění vzdálené kalibrace snímačů využijeme modely *Generic OnOff Client* a *Generic OnOff Server*. První zmíněný bude publikovat do skupiny 0xc200, ze které bude odebírat *Generic OnOff Server* model. Aktivování a deaktivování výstupu, který bude možné připojit na pin *HD*, bude řízeno stavem *Generic OnOff*. Adresování jednotlivých uzlů respektive elementů s využitím publikování není v našem případě možné, jelikož 1 klient publikuje do skupiny, ze které odebírá více serverů. Publikování *Generic OnOff Client* modelu je použitelné pro situaci, kdy chceme zapnout nebo vypnout kalibraci na všech uzlech, které obsahují *Generic OnOff Server* odebírající z adresy 0xc200. Ovládání kalibrace na konkrétním uzlu je řešeno odesláním zprávy *Generic OnOff Set*.

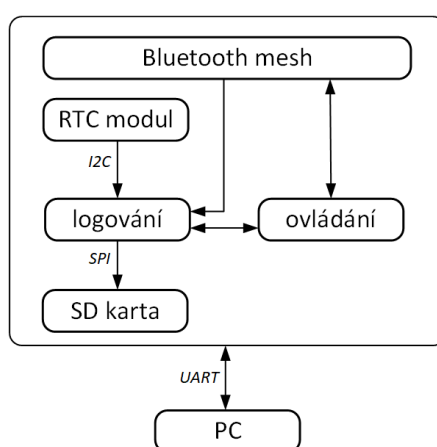


Obr. 5.5: Použité modely a konfigurace adres pro publikování a odebírání zpráv

5.3 Koncentrátor

Hlavním prvkem celého systému je koncentrátor zastávající roli provisionera. Kromě správy nad mesh sítí má za úkol ukládání naměřených dat, které přijal z mesh sítě. Pro koncentrátor je použita vývojová deska Nucleo68. Koncentrátor se ovládá skrz sériovou linku UART, která je emulována přes debugger ST-LINK. Nastavení sériové linky je následující:

- Přenosová rychlost: 115 200 baud
- Počet datových bitů: 8
- Počet stop-bitů: 1
- Parita: žádná
- Řízení toku: žádné



Obr. 5.6: Blokové schéma koncentrátoru

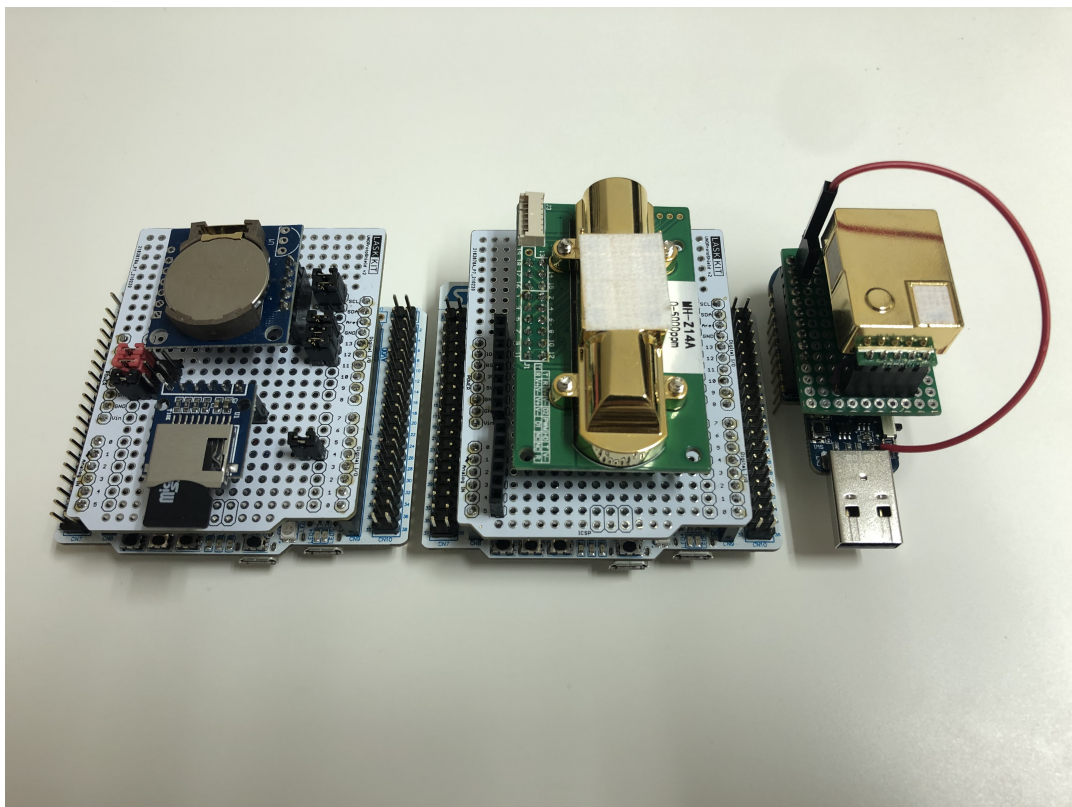
Pro logování dat je použit modul na micro SD kartu, který je k mikrokontroléru připojen přes rozhraní SPI. K zachycení časové známky ukládaných dat slouží modul reálného času *Tiny RTC*, ze kterého je vyčítáno přes sběrnici I2C.

Tab. 5.1: Zapojení modulů koncentrátoru

Modul	Signál	Port MCU	Pin MCU
Tiny RTC	SCL	B	8
Tiny RTC	SDA	B	6
SD karta	MISO	A	6
SD karta	MOSI	A	7
SD karta	CS	C	10
SD karta	SCK	A	5

5.4 Měřicí bod

Měřicí body jsou realizovány na deskách Nucleo68 a WB55 USB Dongle. Analogový výstup snímače je v obou případech připojen na pin A1, který je vnitřním multiplexerem propojen na kanál 6 A/D převodníku. Výstupní signál použitelný pro kalibraci snímače není připojen na vstup snímače HD. Pro demonstraci vzdálené kalibrace je použita zelená LED dioda na vývojových deskách.



Obr. 5.7: Koncentrátor (vlevo) a měřicí body

6 Realizace měřicího systému

Jak již bylo nastíněno v předešlých kapitolách, pro realizaci bezdrátového měřicího systému byly použity platformy od společností STMicroelectronics a Nordic Semiconductor. Modul od druhé zmíněné byl začleněn jen okrajově. Důvod si vysvětlíme později.

Pro STM mikrokontroléry jsme vycházeli z balíčku SDK STM32CubeWB verze 1.10.0, konkrétně ze vzorových příkladů a middleware pro Bluetooth mesh. Používaná verze vývojového prostředí STM32CubeIDE byla 1.4.2. Z počátku vypadalo SDK velice slibně, postupem času jsme však narazili na několik nesprávně fungujících částí v kódu. Některé se podařilo opravit a uvést do funkčního stavu, jiné zase ne. Dokumentace a podpora pro Bluetooth mesh není ze strany STM nikterak velká. Podařilo se dohledat pouze 1 aplikační notu[22], která popisuje Bluetooth mesh z SDK. Kladnou stránkou Bluetooth mesh z SDK je přítomnost některých Bluetooth SIG modelů. Pro modely je předchystána aplikační část, která je pomocí middleware souborů propojena s knihovnou *BLE Mesh*.

6.1 Koncentrátor

Koncentrátor bude zároveň embedded provisioner. Bude obsahovat pouze 1 element, na němž budou implementovány 3 modely, se kterými budeme pracovat. Projekt pro koncentrátor nalezneme v příloze A. V nastavení *mesh_cfg_usr.h* zkontrolujeme, jestli máme definovány následující makra a případně je upravíme.

Výpis 6.1: Konfigurace koncentrátoru

```
1 #define DYNAMIC_PROVISIONER
2 #define ENABLE_PB_ADV
3 #define ENABLE_SERIAL_INTERFACE 1
4 #define ENABLE_SERIAL_PRVN 1
5 #define ENABLE_GENERIC_MODEL_CLIENT_ONOFF (1)
6 #define ENABLE_SENSOR_MODEL_CLIENT (1)
7 #define ENABLE_CONFIG_MODEL_CLIENT (1)
8 #define GENERIC_CLIENT_MODEL_PUBLISH
9
10 #define TTL_DEFAULT 3
11 #define NODE_TRANSMIT_DEFAULT 3
12 #define TR_GAP_BETWEEN_TRANSMISSION 50U
13 #define RELAY_RETRANSMIT_DEFAULT 3 /* only for relay node */
```

První makro povoluje používat uzel jako embedded provisionera. Druhé makro udává

vrstvu, skrz kterou uzel komunikuje, v tomto případě se jedná o *advertising bearer*. Další 2 makra nám umožní ovládat uzel přes sériovou linku. Makra na řádcích 5 až 8 se vztahují k použitým modelům. Zbývající makra se týkají vysílání zpráv na fyzické vrstvě BLE. `TTL_DEFAULT` definuje maximální počet přeskoků jedné zprávy. Kolikrát je zpráva odeslána udává `NODE_TRANSMIT_DEFAULT` a rozestup v jednotkách ms mezi jednotlivými odesláními nastavíme pomocí makra `TR_GAP_BETWEEN_TRANSMISSION`. Délku rozestupu není možné měnit za běhu programu. Pokud bychom přidali uzlu funkci relay, posledním makrem uvedeným ve výpise výše bychom definovali, kolikrát relay uzel přepośle zprávu, kterou přijal od okolních uzlů. Jednotlivé funkce uzlů se definují taktéž v `mesh_cfg_usr.h` a mají koncovku `FEATURE`.

K inicializaci middleware a periférií dochází ve funkci `APPE_Init` (`app_entry.c`). Zde jsme doplnili těla funkcí pro inicializaci I2C, SPI, časovače TIM2 a dalšího výstupního signálu, který bude spolu s TIM2 použit pro měření doby propagace datové zprávy. Funkce uvedené ve výpise 6.3 jsou součástí inicializační rutiny `MESH_Init` (`mesh.c`). Po vykonání všech inicializačních funkcí je z nekonečně smyčky `while{1}` volána funkce `UTIL_SEQ_Run`, ve které je plánováno spouštění úloh registrovaných při inicializaci.

Výpis 6.2: Inicializace I2C, SPI, TIM2 a GPIO

```

1  MX_I2C1_Init();          /* Init I2C for RTC module */
2  SDCard_Log_Init();       /* Init SPI for SD card */
3  LpTimerInit();
4
5  #ifdef MEASURE_MSG_TIME_PROPAGATION
6  OutputSync_Init();       /* Init GPIO synchronization signal */
7  MX_TIM2_Init();          /* Init timer TIM2 */
8  #endif

```

Výpis 6.3: Inicializace parametrů komunikace

```

1  BLEMesh_SetTTL(TTL_DEFAULT);
2  BLEMesh_SetNetworkTransmitCount(NODE_TRANSMIT_DEFAULT);
3  BLEMesh_SetRelayRetransmitCount(RELAY_RETRANSMIT_DEFAULT);

```

6.1.1 Přidání zařízení do sítě

Pro upřesnění nejdříve uvedeme, že přidáním zařízení do sítě je v celé kapitole Realizace měřicího systému myšlen proces provisioning včetně konfigurace modelů, TTL a dalších. Přidání zařízení do sítě mesh bylo v SDK implementováno. Do doby, než byl

koncentrátor restartován, vypadalo, že všechno funguje správně. Po restartu koncentrátoru a následném pokusu o přidání dalšího zařízení do sítě byla tomuto zařízení přidělena adresa 0x0000. Ta je mimo jiné standardem Bluetooth mesh definována jako *nepřiřazená*.

Embedded provisioner si pro každé zařízení, které do sítě přidá, ukládá klíč zařízení, adresu přidělenou 1. elementu a celkový počet elementů obsažených v zařízení. V původní implementaci SDK zabíraly tyto údaje 20 bytů (první 4 byty adresa a počet elementů, zbylých 16 bytů klíč). Tato velikost definuje velikost 1 stránky, která je vyhrazena pro 1 uzel v nevolatilní paměti flash (dále jen NVM). Oblast pro zapisování dat provisionera začíná na adrese 0x80C7000 a má velikost 4096 bytů. Do NVM se zapisuje po 8 bytech. Po vytvoření sítě a přidání 1 zařízení do sítě vypadala NVM následovně:

Address	0 - 3	4 - 7	8 - B	C - F
080C7000	01000101	A5840CE1	761C0D0E	5578BA44
080C7010	D808EC59	00000001	FFFFFFFF	FFFFFFFF
080C7020	FFFFFFFF	FFFFFFFF	02000102	7256B873
080C7030	79F87F51	23E506E9	03557B4F	00000001
080C7040	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF
080C7050	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF

Obr. 6.1: Obsah NVM při 2 uzlech v síti mesh (před úpravou)

Po restartu MCU se hledá první volná stránka, kterou značí hexadecimální hodnota 0xFFFFFFFF na začátku stránky. Ta je nalezena na adrese 0x80C7050. Pro zjištění adresy posledního přidaného zařízení se v paměti podíváme o 1 stránku v paměti zpět, tedy na adresu 0x80C703C. První 2 byty značí adresu ve tvaru big-endian. Je zřejmé, že adresa 0x0000 není platná. Pro správné zapisování a vyčítání dat provisionera musíme změnit velikost stránky tak, aby odpovídala skutečnému počtu zapsaných bytů do NVM. Úpravu provedeme v *appli_nvm.h*:

Výpis 6.4: Změna velikosti stránky pro data provisionera

```
1 #define PRVN_NVM_CHUNK_SIZE          (sizeof(NvmPerNode_t)+4)
2 #define PRVN_NVM_SUBPAGE_SIZE        (sizeof(NvmPerNode_t)+4)
```

Po této úpravě má stránka velikost 24 bytů a dochází již ke korektnímu čtení a zápisu dat. Při zápisu po 8 bytech má NVM následující obsah:

0x80c7000 : 0x80C7000 <Hex> New Renderings...				
Address	0 - 3	4 - 7	8 - B	C - F
080C7000	01000101	A247137D	769AFCF2	FEA197F2
080C7010	1059D8C0	00000000	02000102	3A4DD3F5
080C7020	074B0EC7	F9A3D77B	0FF7494C	00000000
080C7030	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF
080C7040	FFFFFFFF	FFFFFFFF	FFFFFFFF	FFFFFFFF

Obr. 6.2: Obsah NVM při 2 uzlech v síti mesh (po úpravě)

Pro lepší správu mesh sítě si ke každému uzlu ukládáme do NVM informaci, jestli je stále součástí sítě či nikoliv. Velikost stránky je v konečné formě 32 bytů a teoreticky může naše síť obsahovat 128 uzlů. Aplikační nota [22] uvádí, že je možné do sítě mesh přidat zařízení, které je mimo dosah provisionera, a to za účasti dalších uzlů sítě. Tato funkce byla testována, ale funkčnost se nepotvrdila.

Problémy při přidání zařízení do sítě

Přidat zařízení do sítě se ne vždy úspěšně podaří. Můžou nastat tyto situace:

- Nastane chyba v průběhu provisioningu - V tomto případě je uživatel pouze informován o selhání chybovým hlášením **Provisioning failed**.
- Zařízení se stane provozovaným, ale nedojde ke konfiguraci - Uživatel je informován hlášením **No response from node. Node will be unprovision.** Opětovné spuštění konfigurace bylo vyzkoušeno, bohužel nevedlo k vyřešení problému. Proto je po neúspěšné konfiguraci uzel odebrán ze sítě.

6.1.2 Odebrání zařízení ze sítě

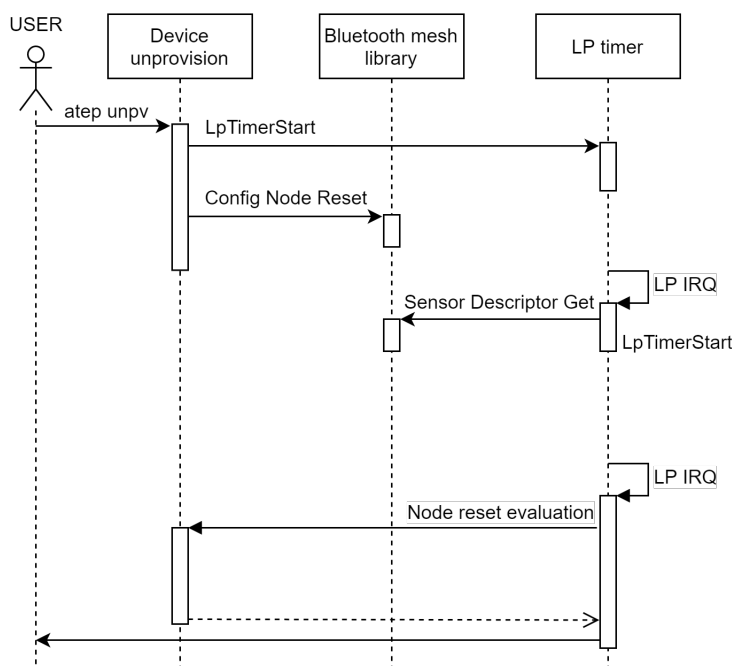
První a zároveň bezpečnější možnost, jak odebrat zařízení ze sítě, je použít tzv. *key refresh*. Princip spočívá ve změně síťového klíče, případně i aplikačních klíčů. Do celého procesu není zahrnut pouze uzel, který má být odebrán ze sítě.

Druhou variantou je odeslat zprávu *Config Node Reset* odebíranému uzlu. Tato varianta nezajistí, že budou skutečně smazány informace o síti na adresovaném uzlu. Implementace našeho systému s touto variantou počítá a při obdržení Config Node reset zprávy dochází ke smazání klíčů a dalších dat. Pro odebrání zařízení je tedy použita druhá zmíněná možnost.

Odebírání zařízení ze sítě nebylo v původním SDK funkční. Při volání knihovni funkce `ConfigModel_SendMessage` z funkce `ConfigClientModel_SendMessage` (`config_client.c`) byl předáván klíč zařízení, které bylo naposledy přidáno do mesh, tudíž

ani nemohla být zpráva doručena požadovanému uzlu. Tento problém nastal i při posílání zpráv pro konfiguraci publikovacích a odebíracích adres. Řešením je vyčíst klíč zařízení náležící cílovému uzlu z oblasti NVM, která začíná na adrese 0x80C7000.

Zpráva Config Node Reset je potvrzovaná a odpověď na ní je *Config Node Reset Status*. Pokud dorazí potvrzovací zpráva, knihovna by měla vyvolat callback funkci `ConfigClientModel_ProcessMessageCb` (`config_client.c`), do které je předán operační kód 0x804A. Ten odpovídá *Config Node Reset Status* zprávě. Při odebrání zařízení bohužel nedochází k vyvolání callback funkce, čímž ztrácíme zpětnou vazbu o stavu uzlu. Na obr. 6.3 je nastíněno implementované řešení. Po požadavku na odebrání zařízení je spuštěn časovač s periodou 1000 ms a v zápětí je mu odeslána *Config Node Reset* zpráva. Po uplynutí časovače je na stejný uzel odeslána *Sensor Descriptor Get* zpráva a opět je spuštěn časovač. Pokud před uplynutím časovače dojde *Sensor Descriptor Status* zpráva, uložíme si příznak, který značí neúspěšné odebrání zařízení ze sítě. Po vypršení časovače je vyhodnoceno, zda-li bylo zařízení úspěšně odebráno ze sítě a uživatel dostává zpětnou vazbu.



Obr. 6.3: Odebrání zařízení ze sítě mesh

Problémy při odebrání zařízení ze sítě

Může nastat situace, že odebírané zařízení nepřijme zprávu Config Node Reset. Vzápětí však obdrží zprávu *Sensor Descriptor Get*, na kterou odpoví. Pokud není odpověď doručena provisioneru, je odebrání zařízení vyhodnoceno jako úspěšné. To samé

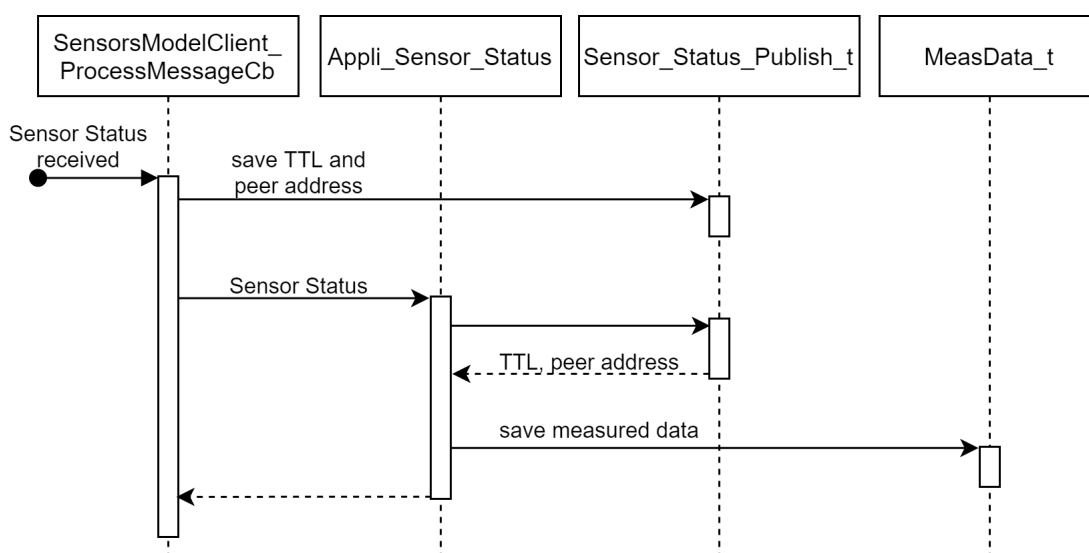
platí, pokud je odebírané zařízení odpojeno od napájení. V prvním případě můžeme zkontrolovat, jestli uzel stále publikuje. Pokud ano, odebrání zařízení zopakujeme.

6.1.3 Přidání a odebrání publish/subscribe adres

Výchozí verze SDK obsahovala statické přidělení adres pro publikování a odebírání zpráv. Pro dynamické nastavování těchto adres byly vytvořeny příkazy, které jsou k dispozici pro uživatele. Jejich používání bude popsáno později. Pro nastavení adres jsou použity zprávy *Config Model Publication Set*, *Config Model Subscription Add* a *Config Model Subscription Delete*. Všechny zprávy jsou potvrzované od cílového uzlu. Bohužel, k vyvolání callback funkce `ConfigClientModel_ProcessMessageCb`, ze které se mimo jiné volají status zprávy pro přidání/odebrání adres, dochází jen do chvíle, kdy je koncentrátor restartován, respektive odpojen od napájení. Poté sice přidání a odebrání adres funguje korektně, ale chybí nám potvrzení o výsledku nastavení.

6.1.4 Příjem dat ze senzorů

Příjem a ukládání měřených dat zobrazuje diagram na obr. 6.4. Po obdržení *Sensor Status* zprávy je knihovnou vyvolána funkce `SensorModelClient_ProcessMessageCb`. Pro přiřazení měřených dat správnému uzlu je zaznamenána adresa. Ukládání TTL bylo využito při měření parametrů sítě. Následuje volání `Appli_Sensor_Status`, kde jsou měřená data uložena, respektive akumulována do struktury `MeasData_t`.



Obr. 6.4: Příjem a ukládání měřených dat

Výpis 6.5: Definice struktury MeasDatat

```

1 typedef struct{
2   uint16_t ui16NodeAdr;    /* Node address */
3   uint32_t ui32AccMeas;    /* Accumulated measured data */
4   uint8_t ui8CounterMeas; /* Number of accumulated samples */
5   uint32_t ui32AvgMeas;    /* Average of accumulated data */
6   uint16_t ui16PropertyID; /* Property ID of sensor descriptor */
7   /* Disable/enable print measured values to terminal */
8   uint8_t bPrintMeasValueEnable;
9   /* Disable/enable log measured data */
10  uint8_t bLogMeasValueEnable;
11 } MeasData_t;

```

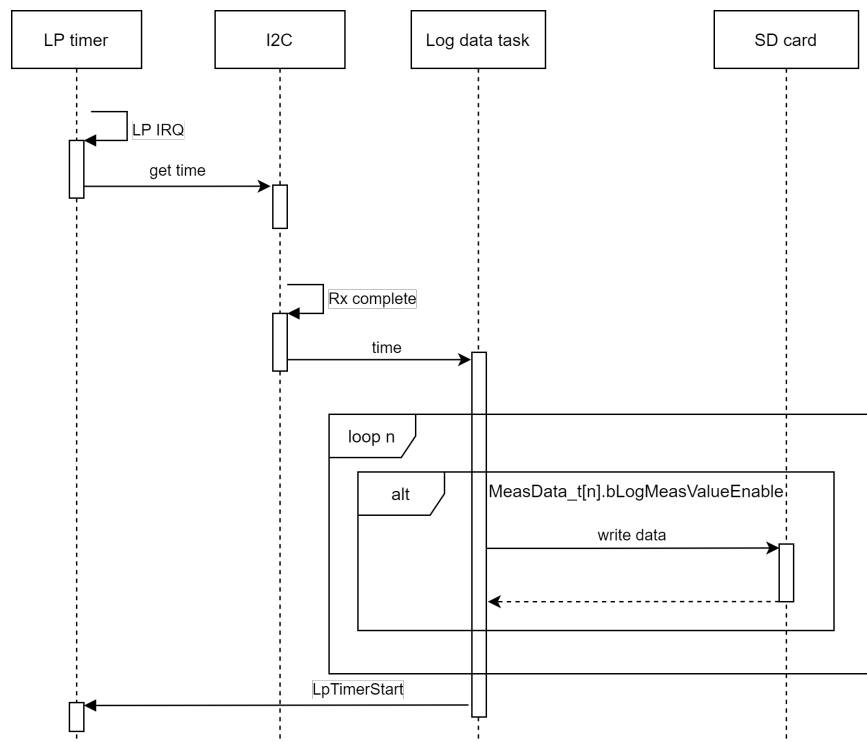
6.1.5 Logování měřených hodnot

Před spuštěním ukládání měřených hodnot na SD kartu se zaznamená, které uzly jsou v síti. Následuje spuštění časovače, po jehož uplynutí se přečte čas z modulu Tiny RTC. V poslední fázi dochází k ukládání dat, které probíhá pro všechny uzly, i pokud uzel nepublikuje. V tomto případě je zaznamenaná hodnota rovna 0. Pro publikující uzly je výsledná uložená hodnota pouhým průměrem z hodnot akumulovaných od předešlého logování.

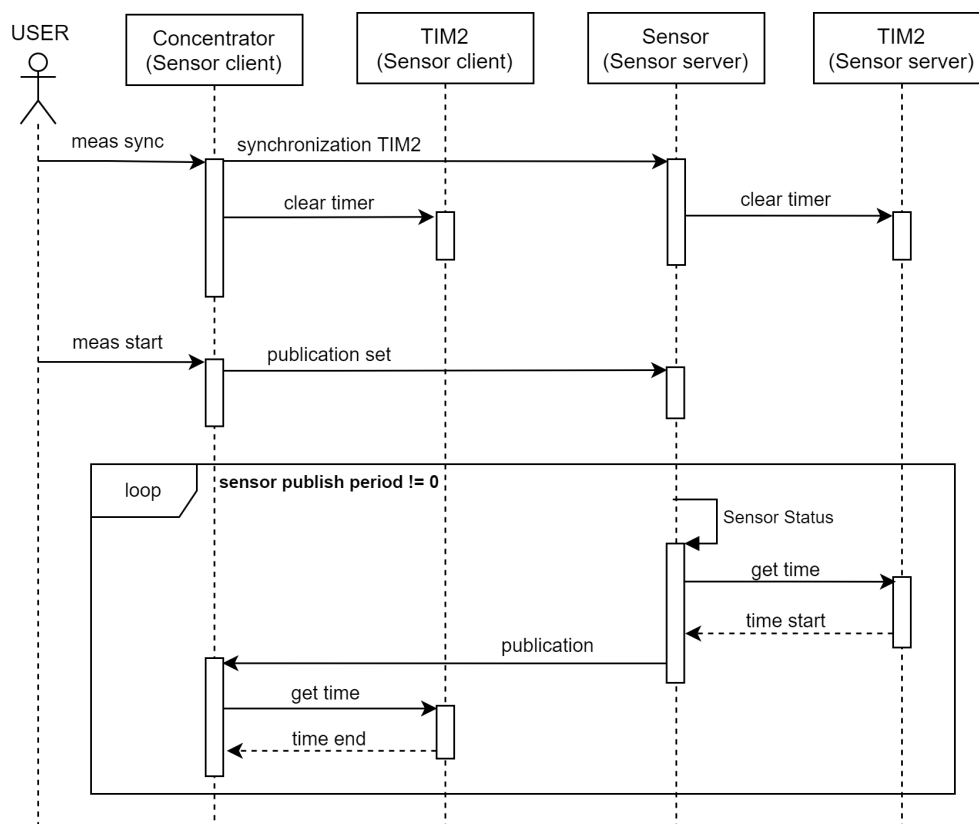
Po přidání nového uzlu do sítě je nutné logování zastavit a znovu spustit. Každý uzel má vlastní logovací soubor. Název souboru odpovídá mesh adrese uzlu. Formát uložených dat je následující: YYYY/MM/DD HH:MM:SS PropertyID MeasuredValue. Např. měřená hodnota koncentrace CO₂ 454 ppm bude uložena jako: 2021/05/01 08:21:43 0x0077 00454.

6.1.6 Měření latence propagace datové zprávy

Pro měření latence byl použit jednoduchý princip, který i přes svá úskalí cíl splnil. Na obr. 6.6 je naznačen princip měření. Uživatel zadá příkaz pro synchronizaci časovačů TIM2 na koncentrátoru a měřicím bodě, které mají rozlišení 100 µs. Synchronizace probíhá přes vodič, který je na koncentrátoru připojen na pin A9 a na měřicím bodu na pin A8. Při publikování na straně *Sensor serveru* a při přijetí Sensor Status zprávy na straně *Sensor client* ukládáme hodnoty časovačů, ze kterých vypočítáme latenci propagace datové zprávy jako: $latence = (time\ end) - (time\ start)$. Měření se povoluje v app_conf.h makrem MEASURE_MSG_TIME_PROPAGATION a je plně implementováno pouze na Nucleo68.



Obr. 6.5: Logování měřených dat



Obr. 6.6: Měření latence propagace datové zprávy

6.1.7 Ovládání aplikace

Nyní si představíme příkazy, kterými je koncentrátor ovládán.

Výchozí příkazy

- `atep root` – Vytvoří mesh síť.
- `atep scan` – Skenování neprovozovaných zařízení v okolí provisionera.
- `atep reset` – Restart zařízení.
- `atep prvn-Dev` – Zahájení provisioningu zařízení. Dev odpovídá zařízení s konkrétním UUID vylistovaným po skenování.
- `atep unpv NodeAdr` – Odebrání uzlu s adresou `NodeAdr` ze sítě mesh.
- `atut set-2` – Smazání mesh sítě. Síť je zrušena pouze na straně embedded provisionera, ostatní uzly v nejsem odebrány.

Vlastní příkazy

- `log en period` – Povolení logování dat. Perioda logování `period` [s].
- `log dis` – Zakázání logování dat.
- `pub set NodeAdr PubAdr ModelID period` – Nastavení publikování s periodou `period` [s].
 - `NodeAdr` – Adresa cílového uzlu.
 - `PubAdr` – Adresa publikování.
 - `ModelID` – Identifikační číslo modelu [20].
- `sub del NodeAdr SubAdr ModelID` – Smazání adresy pro odebírání zpráv.
 - `NodeAdr`, `ModelID` – Viz předchozí.
 - `SubAdr` – Adresa pro odebírání.
- `sub add NodeAdr SubAdr ModelID` – Přidání adresy pro odebírání zpráv.
 - `NodeAdr`, `ModelID`, `SubAdr` – Viz předchozí.
- `print on NodeAdr` – Povolení zobrazování měřených hodnot z uzlu `NodeAdr`.
- `print off NodeAdr` – Zakázání zobrazování měřených hodnot z uzlu `NodeAdr`.
- `meas sync` – Synchronizace časovačů TIM2.
- `meas start NodeAdr period` – Spuštění měření. Dojde k nastavení publikování s periodou `period` [s] na uzlu `NodeAdr`. Adresa publikování a identifikační číslo modelu jsou výchozí.
- `meas stop NodeAdr` – Zastavení publikování uzlu `NodeAdr`.
- `calib on NodeAdr` – Nastaví stav Generic OnOff na vzdáleném uzlu `NodeAdr` na logickou hodnotu 1.
- `calib off NodeAdr` – Nastaví stav Generic OnOff na vzdáleném uzlu `NodeAdr` na logickou hodnotu 0.
- `atep printnd` – Výpis provozovaných zařízení v síti mesh.

6.2 Měřicí bod

Měřicí bod bude obsahovat 1 element, na němž budou umístěny 2 modely, se kterými budeme pracovat. Projekty pro měřicí body nalezneme v příloze A. Opět zkontrolujeme nastavení v *mesh_cfg_usr.h* tak, aby odpovídalo výpisu 6.6. Konfiguraci vztahující se k Sensor Server modelu najdeme v *sensor_cfg_usr.h*. Zde se nastavuje PropertyID senzoru, rozsah, přípustné tolerance a další.

Výpis 6.6: Konfigurace měřicího bodu

```
1 #define ENABLE_PB_ADV
2 #define ENABLE_GENERIC_MODEL_SERVER_ONOFF      (1)
3 #define ENABLE_SENSOR_MODEL_SERVER             (1)
4 #define GENERIC_SERVER_MODEL_PUBLISH
5 #define ENABLE_SENSOR_PUBLICATION
6 #define NUMBER_OF_SENSOR                       1
7 #define TTL_DEFAULT                           3
8 #define NODE_TRANSMIT_DEFAULT                 3
9 #define TR_GAP_BETWEEN_TRANSMISSION           50U
10 #define RELAY_RETRANSMIT_DEFAULT 3 /* only for relay node */
```

Data jsou ze senzoru vyčítána s periodou 1 sekundy. Podle zvoleného rozsahu, v našem případě 10 000 ppm, je po dokončení A/D převodu proveden přepočít na měřenou koncentraci. Pokud Sensor Server model publikuje, měřená hodnota je ve funkci **Appli_Sensor_ReadValue** (*appli_sensor.c*) předána do middleware vrstvy, která se postará o odeslání dat.

Narazily jsme na problém, že po restartu zařízení nedochází k periodickému publikování. Problém byl vyřešen tak, že je po restartu nastavena výchozí publikovací perioda ve funkci **Mesh_Init** (*mesh.c*).

V kapitole 5.2.1 byla zmíněná možnost vzdálené kalibrace, která spočívá v přivedení napětí 0V na vstup senzoru HD. Pro demonstraci vzdálené kalibrace budeme rozsvěcovat a zhaset zelenou LED diodu. Příkazem pro kalibraci je na měřicím bodě vyvolána funkce **Appli_Generic_OnOff_Set** (*appli_generic.c*).

Výpis 6.7: Použití stavu Generic OnOff pro vzdálenou kalibraci

```
1 if((AppliOnOffSet[elementIndex].Present_OnOff == APPLI_LED_ON)
2 && (OptionalValid == NO_TRANSITION))
3 {   BSP_LED_On(LED_GREEN);   /* Calib On */ }
4 else
5 {   BSP_LED_Off(LED_GREEN);  /* Calib Off */ }
```

6.3 Začlenění nRF52840 USB Dongle

Modul nRF52840 USB Dongle byl při realizaci měřicího systému používán minimálně, zejména jen pro větší vytížení mesh sítě. Hlavním důvodem byla absence debuggeru. Podpora nRF52840 USB Dongle je omezená. Také na stránkách výrobce je pro účely vývoje doporučovaná verze nRF52840 DK, která obsahuje debugger[30].

Alespoň okrajově se podíváme na příklad z SDK Mesh, který po stažení SDK nalezneme v `..src\examples\sensor\server`. Volíme projekt pro mikrokontroléry řady *nrf52840*. Nejdříve musíme změnit desku, pro kterou bude program přeložen. Otevřeme nastavení projektu (pravým tlačítkem myši - Options) , vybereme konfiguraci *Common* a aktivujeme záložku *Preprocessor*. Dále otevřeme *Preprocessor Definitions*, kde změníme `BOARD_PCA10056` na `BOARD_PCA10059` a uložíme. V souboru `main.c` je vytvořena struktura `m_sensor_server_0` sloužící pro Sensor Server model. Velmi přehledně jsou zde mapovány callback funkce volané při různých událostech mesh sítě. Data pro publikování jsou předávány v callback funkci `app_sensor_get_cb`. Konfiguraci některých parametrů mesh je možné měnit v `nrf_mesh_config_core.h`. Nastavují se zde např. typy uzlů (relay, LPN, friend), počet odeslání zprávy nebo zde můžeme povolit obnovení publikovací periody, což u STM schází.

Vzorový příklad jsem se pokusil upravit tak, aby Sensor Server odpovídal měření koncentrace CO₂. V `main.c` bylo změněno `PropertyID` senzoru na `0x0077`. Dále v `app_sensor_utils.c` ve funkcích `range_vector_bytes_get`, `delta_vector_bytes_get`, `sensor_initialize` byly přidány instance case pro `propertyID = 0x0077`. Po přidání zařízení do sítě Sensor Server publikoval, ale na Sensor Client modelu docházelo k vícenásobnému volání funkce pro Sensor Status. To i přes to, že byla callback funkce `app_sensor_get_cb` volána se správnou periodou, což bylo ověřeno blikáním LED diody. Kvůli absenci debuggeru se nepodařilo odhalit, kde může vznikat chyba.

6.4 Odebrání zařízení ze sítě pomocí uživatelských tlačítek

Platí pouze v rámci uzlu, na kterém se odebrání provádí. Provisioner nedostane informaci, že byl uzel odebrán někým jiným.

STM

V případě Nucleo68 platí tato sekvence:

- Stiskneme Reset (SW4), poté stiskneme SW1.
- Uvolníme Reset a SW1 držíme do doby, než se rozbliká modrá LED dioda. Poté je zařízení neprovozované a může být opět přidáno do mesh sítě.

Pro odebrání WB55 USB Dongle stiskneme před připojením napájení SW1 a držíme do doby, než se rozbliká modrá LED dioda.

Nordic Semiconductor

V main.c je třeba upravit funkci `button_event_handler` viz obr. 6.7. Po stisknutí a podržení SW1 dojde k vymazání údajů o mesh síti.

```
static void button_event_handler(uint32_t button_number)
{
    timestamp_t m_last_button_press; /* Time */
    button_number++; /* Increase to match number printed on DK */
    __LOG(LOG_SRC_APP, LOG_LEVEL_INFO, "Button %u pressed\n", button_number);
    switch (button_number)
    {
        case 1:
        {
            m_pir_motion_sensed_in_period = (m_pir_motion_sensed_in_period > 1)
                                            ? m_pir_motion_sensed_in_period - 1 : 0;
            timestamp_t m_last_button_press = timer_now();
            while(!nrf_gpio_pin_read(BUTTON_1));
            if (TIMER_DIFF(m_last_button_press, timer_now()) > MS_TO_US(3000))
            {
                mesh_stack_config_clear();
                node_reset();
            }
            break;
        }
    }
}
```

Obr. 6.7: Odebrání nRF52840 USB Dongle pomocí SW1

6.5 Komplikace, které se nepovedlo vyřešit

Přímý přístup k měřicímu zařízení

Jak již bylo zmíněno, pomocí GATT bearer můžou s uzlem mesh komunikovat zařízení, které nepodporují Bluetooth mesh. Při provisioningu se používá služba *Mesh Provisioning* (UUID = 0x1827), poté služba *Mesh Proxy* (0x1828). Po povolení relay funkce a GATT bearer je zařízení viditelné v mobilní aplikaci nRF Connect. Po provisioningu u tohoto zařízení vidíme, že disponuje službou Mesh Proxy. Po připojení k tomuto zařízení můžeme vidět i charakteristiky *Mesh Proxy Data In* a *Mesh Proxy Data Out*. Čtení, případně zápis do těchto charakteristik nevyvolá žádnou akci na uzlu. V SDK (STM) ani v aplikační notě AN5292[22] není k této problematice žádná zmínka vyjma toho, kde povolit/zakázat proxy uzel.

Ověření Output OOB

Pro zopakování, k ověření zařízení dochází při provisioningu. Při metodě Output OOB signalizuje neprovozované zařízení náhodné číslo, které uživatel zadá do provisionera. Podporu této metody ověření povolíme v `mesh_cfg_usr.h`. Dle dostupných

zdrojových kódů a komentářů v SDK by mělo při provisioningu dojít na neprovozaném zařízení k vyvolání funkce `Appli_BleOutputOOBAuthCb` (`appli_mesh.c`). Ta by pravděpodobně předala náhodné číslo, podle kterého by se rozblikala LED dioda. Žádný další postup se bohužel z kódu nedá určit. Aplikační nota AN5292 tuto metodu taktéž nijak nepopisuje.

Selhání sériové linky

V průběhu práce jsme několikrát narazili na problém, že občas přestane komunikovat sériová linka. Při krokování programu nebyla nalezena chyba. Program běžel bez problémů, pouze přestalo být voláno přerušování od sériové linky. Mimo jiné, kvůli tomuto problému byly použité snímače připojeny pouze na A/D převodník, nikoliv na LPUART, která byla původně zamýšlena.

Selhání vysílání mesh zpráv

Obsah zprávy Sensor Status nechť jsou užitečná data. Zpráva pro měření koncentrace CO₂ obsahuje 4 byty užitečných dat. Při měření délky propagace datové zprávy byla velikost užitečných dat *uměle* zvětšována. Pro velikost užitečných dat 10 bytů a méně fungovalo vysílání zpráv korektně. Pro větší množství bytů přestává uzel po několika publikování vysílat zprávy. Velikost zprávy by neměla ovlivňovat publikování, pouze bude docházet k větší latenci kvůli segmentaci zprávy [27]. Měření latence a úspěšnosti přenosu zprávy nebylo kvůli tomuto problému změřeno pro zprávy, která obsahují užitečná data o velikosti 11 bytů a více.

7 Naměřené parametry systému

V prostorech popsaných v kapitole 5.1 byly změřeny tyto parametry:

- Úspěšnost přenosu datové zprávy
- Délka propagace datové zprávy (dále jen latence)
- Průměrný odběr proudu zařízení

Pro LPN byl změřen proud ve spánkovém režimu a úspěšnost doručení zprávy.

7.1 Úspěšnost přenosu a latence datové zprávy

Měření prvních dvou parametrů bylo uskutečněno pro několik konfigurací umístění uzlů v síti. Při volbě konfigurace jsme vycházeli z rozmístění uzlů viz obr. 5.1, 5.2, 5.3. Tabulka 7.1 obsahuje vybrané konfigurace. Každá konfigurace byla měřena pro přenášenou velikost užitečných dat (dále jen payload) 7 a 10 bytů. Tabulka 7.2 obsahuje přiřazené adresy a funkce uzlů.

Tab. 7.1: Konfigurace měření

Konfigurace	1	2	3	4
S1	0x2	0x2	0x6	0x6
S2	0x4	0x4	0x4	0x4
S3	0x5	0x5	0x5	-
S4	0x6	0x6	0x2	0x5
S5	0x7	0x7	0x7	0x7
S6	-	-	-	0x2
R1	-	0x3	0x3	0x3
R2	-	-	-	0x8
C	0x1	0x1	0x1	0x1

Tab. 7.2: Přidělené adresy a funkce uzlů

Adresa	Funkce	Komentář
0x1	koncentrátor	-
0x2	koncový bod	použit pro měření latence
0x3, 0x8	relay	-
0x4 – 0x7	koncový bod	-

Měřicí podmínky úspěšnosti a latence zprávy

Měřicí podmínky byly při všech konfiguracích následující:

- TTL: 3
- Počet odeslání 1 mesh zprávy: 3

- Počet přeposlání 1 mesh zprávy relay uzlem: 3
- Rozestup mezi odesláním zpráv: 50 ms
- Vysílací výkon: 0 dBm

V poslední řadě uvedeme zkratky používané v tabulkách níže a vzorce pro výpočet:

- T_P – Perioda publikování.
- Δt_k – Průměrná latence zprávy s k-přeskoky. Při každém měření odesláno přibližně 100 zpráv.
- σ^2_k – Rozptyl latence zprávy s k-přeskoky (vzorec 7.1).
- σ_k – Směrodatná odchylka latence zprávy s k-přeskoky (vzorec 7.2).
- \min_k – Minimální latence zprávy s k-přeskoky.
- \max_k – Maximální latence zprávy s k-přeskoky.

$$\sigma^2 = \frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2 \quad (7.1)$$

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2} \quad (7.2)$$

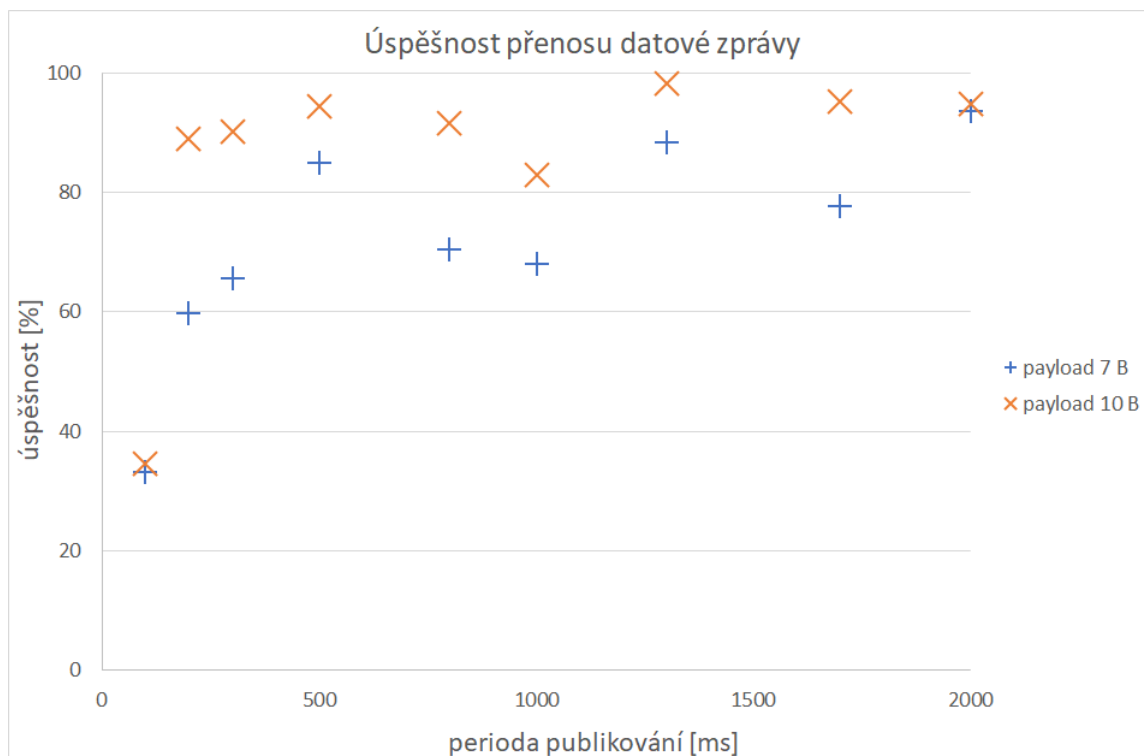
7.1.1 Konfigurace 1

Tab. 7.3: Naměřené hodnoty konfigurace 1, payload 7 bytů

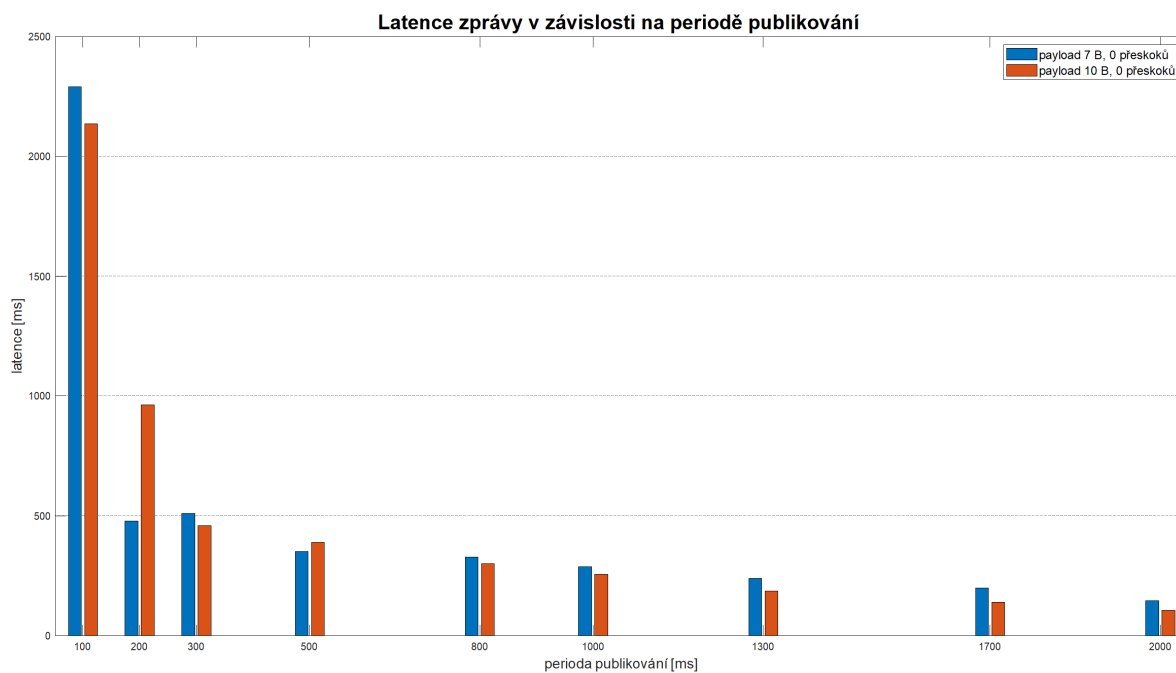
T_P [ms]	2000	1700	1300	1000	800	500	300	200	100
Δt_0 [ms]	145.2	199.0	239.9	287.3	327.3	351.1	510.9	478.1	2291.7
σ^2_0 [ms]	1176.3	1180.8	1428.9	1052.6	1860.4	12161.0	1579.1	2627.8	8715.9
σ_0 [ms]	34.3	34.4	37.8	32.4	43.1	34.9	39.7	51.3	93.4
\min_0 [ms]	117.2	159.7	203.7	247.5	270.7	314.9	467.6	423.5	1811.6
\max_0 [ms]	218.5	260.1	306.7	349.8	407.5	416.8	586.2	654.4	2392.6
úspěšnost [%]	93.5	77.6	88.3	67.9	70.4	84.9	65.6	59.7	33.1

Tab. 7.4: Naměřené hodnoty konfigurace 1, payload 10 bytů

T_P [ms]	2000	1700	1300	1000	800	500	300	200	100
Δt_0 [ms]	105.7	139.4	186.7	255.5	301.1	390.0	459.4	962.4	2136.6
σ^2_0 [ms]	1197.6	975.8	1110.6	1545.3	1173.8	1222.5	2420.3	28356.8	21342.1
σ_0 [ms]	34.6	31.2	33.3	39.3	34.3	35.0	49.2	168.4	146.1
\min_0 [ms]	73.0	116.0	159.9	225.4	268.9	356.7	423.0	602.1	1221.5
\max_0 [ms]	176.6	218.1	294.5	477.0	370.4	457.4	709.2	1238.4	2253.6
úspěšnost [%]	94.8	95.2	98.1	82.9	91.5	94.4	90.1	89.0	34.6



Obr. 7.1: Úspěšnost přenosu datové zprávy, konfigurace 1



Obr. 7.2: Latence zprávy, konfigurace 1

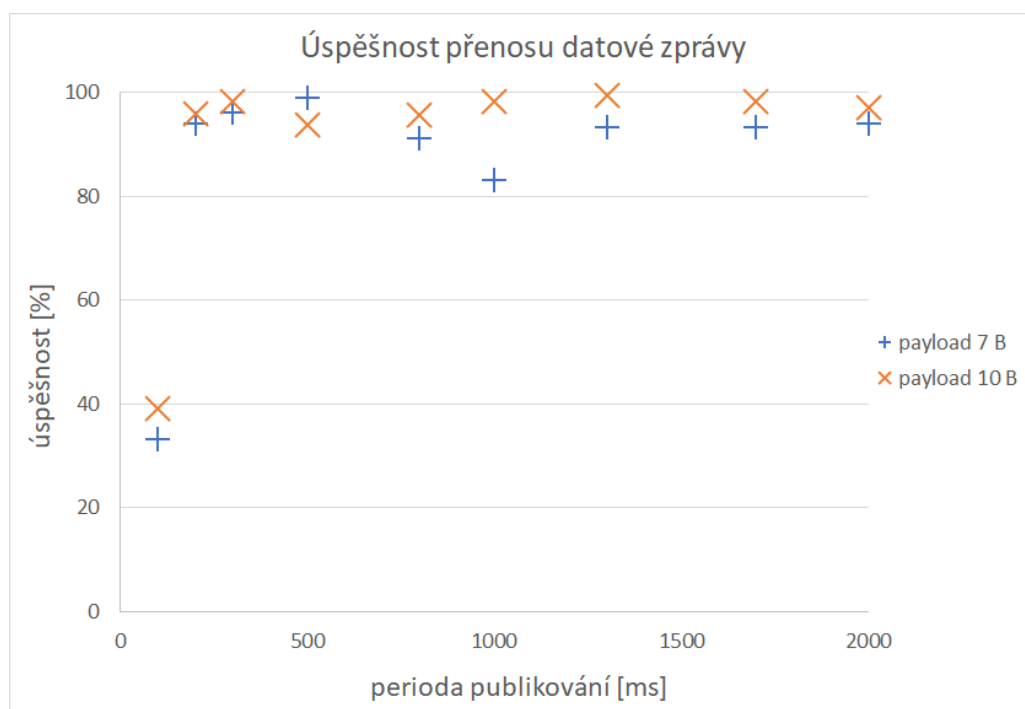
7.1.2 Konfigurace 2

Tab. 7.5: Naměřené hodnoty konfigurace 2, payload 7 bytů

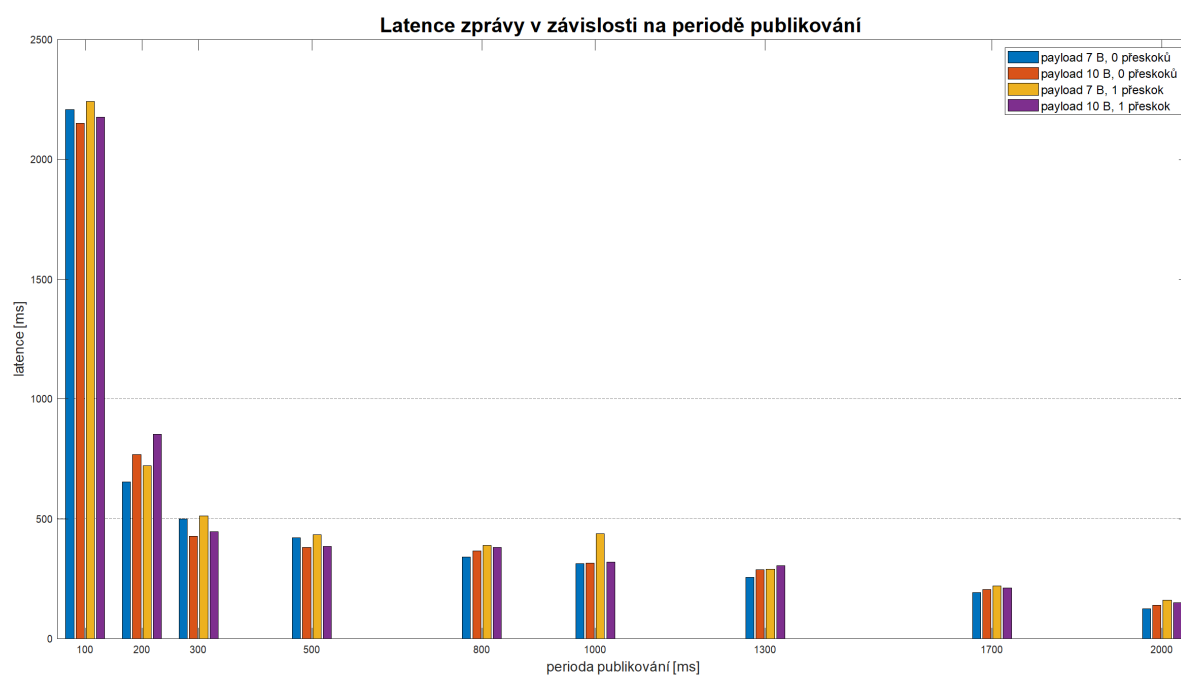
Ts	2000	1700	1300	1000	800	500	300	200	100
Δt_0 [ms]	124.7	192	255	312.7	340.6	421.1	500.2	654.3	2209.1
σ^2_0 [ms]	1079.2	1502.6	1361.5	1496.7	975.1	803	1383.9	16994.1	2643.3
σ_0 [ms]	32.9	38.8	36.9	38.7	31.2	28.3	37.2	130.4	51.4
\min_0 [ms]	93.5	159.3	224.8	268.4	311.7	399	465.4	530.5	2048.1
\max_0 [ms]	196.5	262.9	361.2	371.5	415.9	505.8	608.5	1019.2	2314.8
Δt_1 [ms]	161.4	220.8	289.8	438.5	388.9	434.6	512.5	720.8	2243.1
σ^2_1 [ms]	2523.2	2563	3625.3	2117.2	3436.6	991.1	1975.1	25917.5	6894.4
σ_1 [ms]	50.2	50.6	60.2	46	58.6	31.5	44.4	161	83
\min_1 [ms]	103.6	168.2	233.6	361.1	322.4	408.8	473.8	541.4	1833.8
\max_1 [ms]	301.1	374.1	483.6	521.5	581.3	519.5	674.2	1056.9	2388.8
úspěšnost [%]	93.9	93.3	93.3	83.2	91.2	99	96.2	93.9	33.2

Tab. 7.6: Naměřené hodnoty konfigurace 2, payload 10 bytů

Ts	2000	1700	1300	1000	800	500	300	200	100
Δt_0 [ms]	139.8	205.2	288.8	314.8	366.1	381.4	427.2	766.9	2150.6
σ^2_0 [ms]	885.3	925.1	758.0	1838.1	3563.2	1100.5	979.3	42194.6	2381.7
σ_0 [ms]	29.8	30.4	27.5	42.9	59.7	33.2	31.3	205.4	48.8
\min_0 [ms]	116.0	182.2	269.3	291.8	336.0	357.7	401.9	488.7	1925.1
\max_0 [ms]	216.4	283.6	370.7	571.2	755.8	459.1	525.5	1139.7	2239.0
Δt_1 [ms]	149.9	210.7	304.1	319.4	380.6	384.8	446.5	852.5	2175.9
σ^2_1 [ms]	1112.9	2195.1	910.7	2034.7	1450.6	515.8	2230.8	43262.7	921.8
σ_1 [ms]	33.4	46.9	30.2	45.1	38.1	22.7	47.2	208.0	30.4
\min_1 [ms]	127.1	191.0	278.6	299.5	344.4	364.4	409.1	500.0	2142.8
\max_1 [ms]	255.8	446.1	388.9	564.2	465.4	432.0	593.5	1091.7	2215.4
úspěšnost [%]	97.0	98.2	99.3	98.2	95.7	93.7	98.3	95.8	39.2



Obr. 7.3: Úspěšnost přenosu datové zprávy, konfigurace 2



Obr. 7.4: Latence zprávy, konfigurace 2

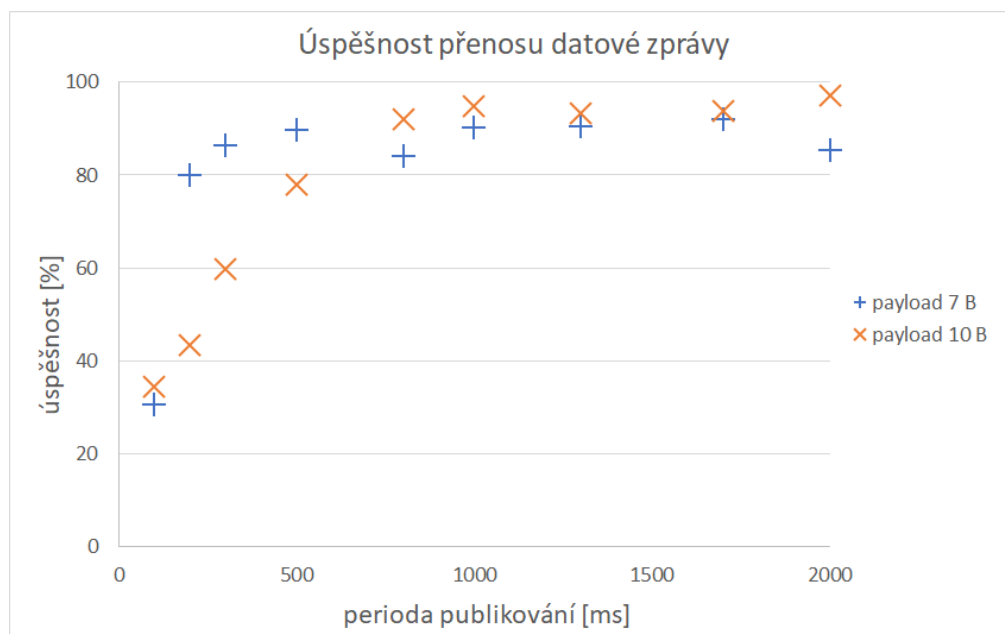
7.1.3 Konfigurace 3

Tab. 7.7: Naměřené hodnoty konfigurace 3, payload 7 bytů

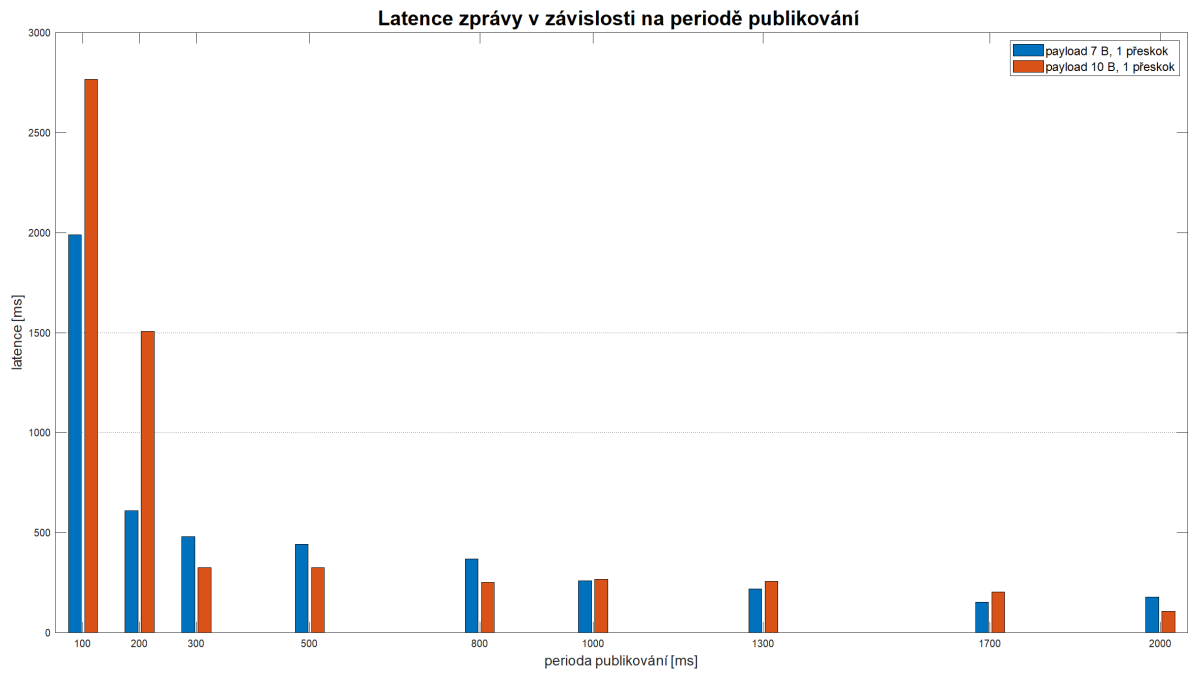
Ts	2000	1700	1300	1000	800	500	300	200	100
Δt_1 [ms]	177	153.4	217.6	259	368.8	441.5	480.9	610	1990.1
σ^2_1 [ms]	2182.7	2589.7	1425.3	824.1	2067.5	2786.4	1779.6	19522.3	9709.8
σ_1 [ms]	46.7	50.9	37.8	28.7	45.5	52.8	42.2	139.7	98.5
\min_1 [ms]	124.6	103.4	169.1	232.4	321.5	386.1	431.6	474.7	1387.4
\max_1 [ms]	309.3	322.6	334.3	347.1	525.4	627.5	628.2	1065.6	2133.8
úspěšnost [%]	85.4	92.1	90.6	90.3	84	89.8	86.3	79.9	30.5

Tab. 7.8: Naměřené hodnoty konfigurace 3, payload 10 bytů

Ts	2000	1700	1300	1000	800	500	300	200	100
Δt_1 [ms]	106.4	202.8	257.6	267.4	251.6	324.8	326.0	1507.2	2766.2
σ^2_1 [ms]	5188.4	6079.8	4637.3	4056.2	7075.3	6616.6	12734.6	79856.6	415121.2
σ_1 [ms]	72.0	78.0	68.1	63.7	84.1	81.3	112.8	282.6	644.3
\min_1 [ms]	37.4	123.8	190.8	213.0	169.1	236.2	235.4	921.5	1262.2
\max_1 [ms]	392.4	524.6	489.1	643.5	563.5	560.1	964.5	1947.6	3546.4
úspěšnost [%]	97.2	93.9	93.2	94.8	92.0	77.9	59.8	43.3	34.5



Obr. 7.5: Úspěšnost přenosu datové zprávy, konfigurace 3



Obr. 7.6: Latence zprávy, konfigurace 3

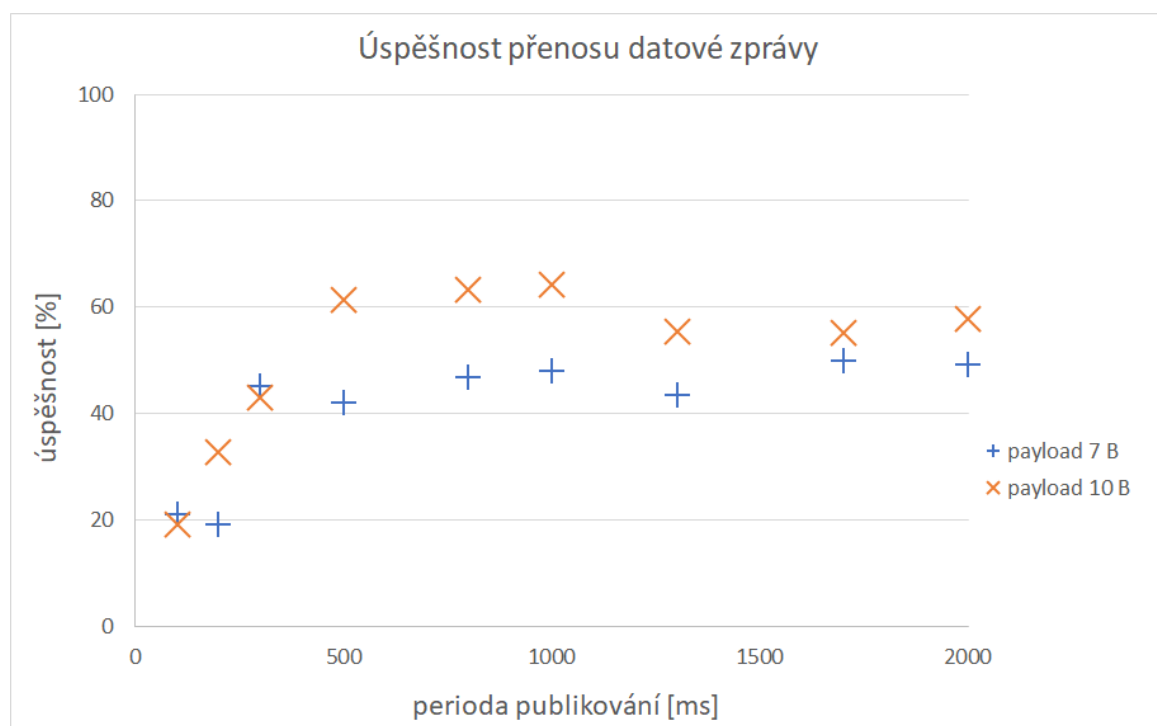
7.1.4 Konfigurace 4

Tab. 7.9: Naměřené hodnoty konfigurace 4, payload 7 bytů

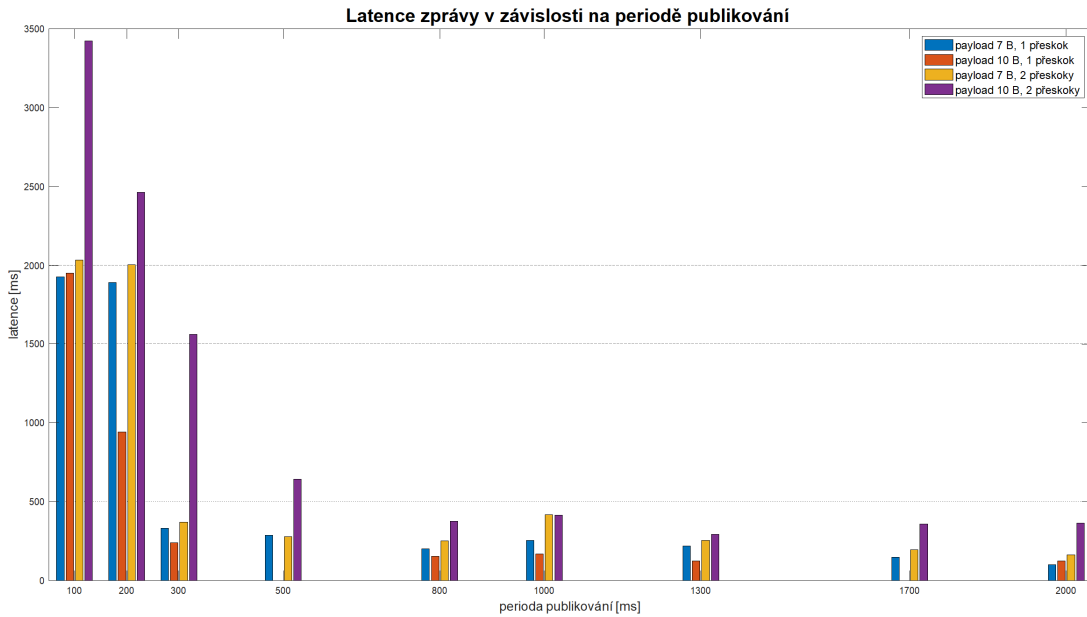
Ts	2000	1700	1300	1000	800	500	300	200	100
Δt_1 [ms]	101.6	148.3	220.3	253.9	201.4	286.3	331.4	1893.3	1927
σ^2_1 [ms]	29.1	297.7	1340	162.9	6811.2	3397.8	12325.8	11665	1926.2
σ_1 [ms]	5.4	17.3	36.6	12.8	82.5	58.3	111	108	43.9
min ₁ [ms]	94.8	119.7	169.8	234.2	150.6	216.8	270.5	1474.2	1855.2
max ₁ [ms]	108.6	192.4	287	300.2	448.9	411	699.5	2025.8	2017.8
Δt_2 [ms]	162.1	195.3	255.3	419	252.2	279.2	371.1	2004.4	2034.4
σ^2_2 [ms]	3459.1	2390.3	4655	1725.9	6883.2	4154.3	6184.1	17039.6	18846.9
σ_2 [ms]	58.8	48.9	68.2	41.5	83	64.5	78.6	130.5	137.3
min ₂ [ms]	109.2	148.9	194.8	381.3	171.6	236.7	280.3	1810.6	1884
max ₂ [ms]	305.7	324.4	532.1	508.1	469.1	541.2	553.7	2326.2	2508.6
úspěšnost [%]	49.2	50	43.5	47.9	46.7	41.9	45.2	19	21

Tab. 7.10: Naměřené hodnoty konfigurace 4, payload 10 bytů

Ts	2000	1700	1300	1000	800	500	300	200	100
Δt_1 [ms]	123.6	0.0	124.5	169.1	152.7	0.0	240.8	942.1	1952.0
σ^2_1 [ms]	12.0	0.0	7.6	78.1	291.2	0.0	167.5	22422.3	972.4
σ_1 [ms]	3.5	0.0	2.8	8.8	17.1	0.0	12.9	149.7	31.2
min ₁ [ms]	120.1	0.0	122.5	157.8	141.4	0.0	232.5	695.3	1930.0
max ₁ [ms]	128.4	0.0	126.4	195.4	172.3	0.0	260.1	1271.0	1974.1
Δt_2 [ms]	365.4	359.8	294.3	414.3	377.7	642.5	1560.8	2465.2	3424.3
σ^2_2 [ms]	40862.5	22573.6	25235.2	17308.5	40304.3	149918.0	35948.1	90589.4	41670.1
σ_2 [ms]	202.1	150.2	158.9	131.6	200.8	387.2	189.6	301.0	204.1
min ₂ [ms]	132.1	194.0	129.3	174.5	153.4	202.2	1210.8	1206.3	2572.7
max ₂ [ms]	845.9	845.2	992.4	763.1	1026.6	1591.8	1943.0	2743.6	3631.8
úspěšnost [%]	57.8	55.1	55.4	64.3	63.3	61.3	43.0	32.7	19.2



Obr. 7.7: Úspěšnost přenosu datové zprávy, konfigurace 4



Obr. 7.8: Latence zprávy, konfigurace 4

7.2 Průměrný odběr proudu zařízení

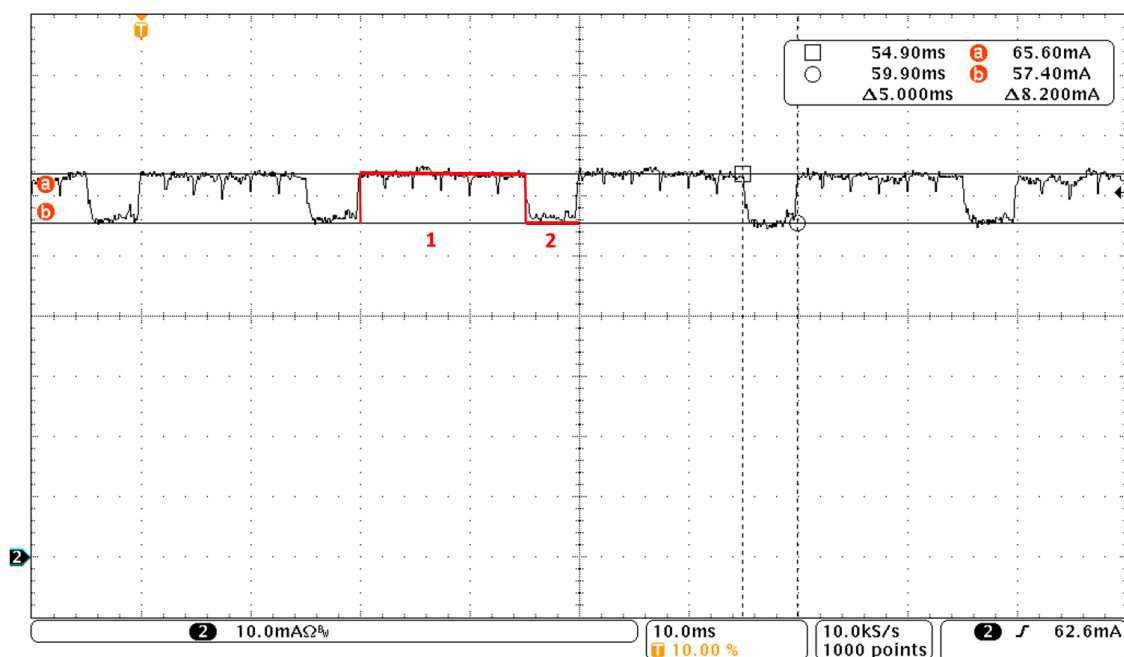
Podmínky při měření proudu byly totožné jako v Měřicí podmínky úspěšnosti a latence zprávy. Pro měření proudu byl použit osciloskop Tektronix MDO4024C s proudovou sondou Tektronix TCP202. Vzorec 7.3 pro výpočet průměrného odběru proudu byl převzat z [31] a upraven. T_S je sledovaná perioda měření, N je počet úseků, na které je perioda měření rozdělena, T_i je délka úseku i a I_i je odběr proudu v úseku i .

S dostupným osciloskopem a proudovou sondou se nepovedlo dostatečně přesně změřit průběhy proudu. Proto budou při výpočtech tyto průběhy aproximovány obdélníkovým signálem. Offset proudu se v průběhu měření měnil, zejména při měření proudu měřicích bodů (Nucleo68 nebo WB55 USB Dongle se senzorem). Posuv offsetu byl až 20 mA.

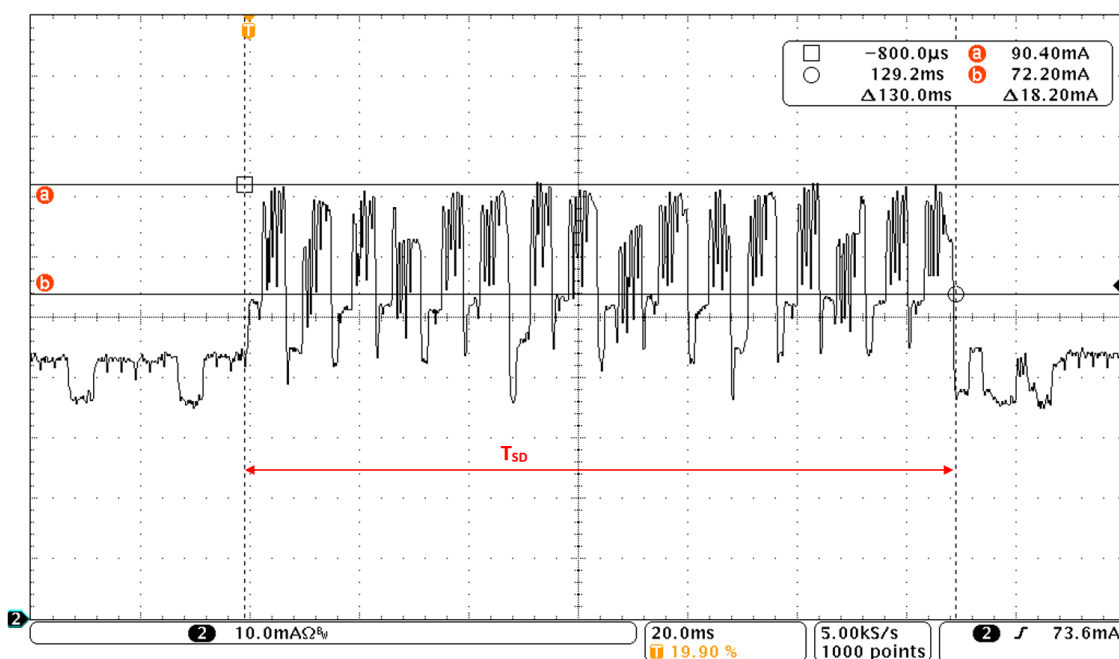
$$I_{AVG} = \frac{1}{T_S} \sum_{i=0}^N T_i \cdot I_i \quad (7.3)$$

7.2.1 Proudový odběr koncentrátoru

Koncentrátor byl při měření napájen přes ST-LINK USB, kde byl také měřen proud. V síti mesh bylo celkem 8 zařízení, z nichž 7 publikovalo s periodou 1 sekunda. Zároveň bylo zapnuté logování dat, jehož perioda byla 10 sekund.



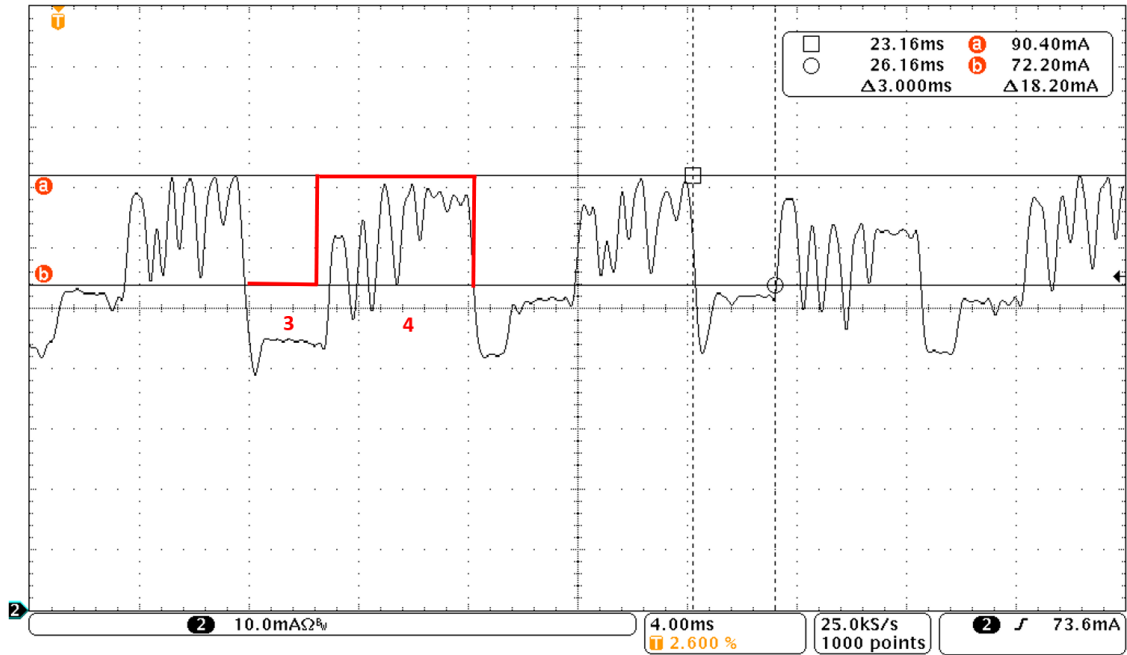
Obr. 7.9: Průběh proudu - koncentrátor



Obr. 7.10: Průběh proudu při zápisu na SD kartu

Na obr. 7.9 vidíme, že odběr proudu není konstantní. Jak již bylo řečeno, pro zjednodušení budeme průběh prokládat obdélníkovým signálem. Z průběhu je také patrné, že se změny proudu periodicky opakují. Budeme předpokládat, že byla střída v průběhu měření konstantní. Tyto předpoklady budeme brát v úvahu i při všech dalších výpočtech.

Při logování dat dochází k nárůstu odběru proudu, jak můžeme vidět na obr. 7.10.



Obr. 7.11: Přiblížení průběhu proudu koncentrátoru při zápisu na SD kartu

Délka zápisu na kartu T_{SD} trvá 130 ms. V tomto úseku můžeme vidět periodicitu. Tuto část si opět proložíme obdélníkovým signálem (viz obr. 7.11), jehož perioda je $130ms/16 = 8.125$ ms. Dále si označíme $T_{MESH} = T_S - T_{SD}$, $T_{12} = T_1 + T_2$.

Tab. 7.11: Odběr proudu koncentrátorem - rozdělení na úseky

Úsek	1	2	3	4
T [ms]	15	5	3	5.125
I [mA]	65.6	57.4	72.2	90.4

Při výpočtu podle vzorce 7.3 nesmíme zapomenout, že se úseky, na které je signál rozdělen, několikrát opakují. Podle periody logování je $T_S = 10$ sec. Průměrný odběr proudu je pak:

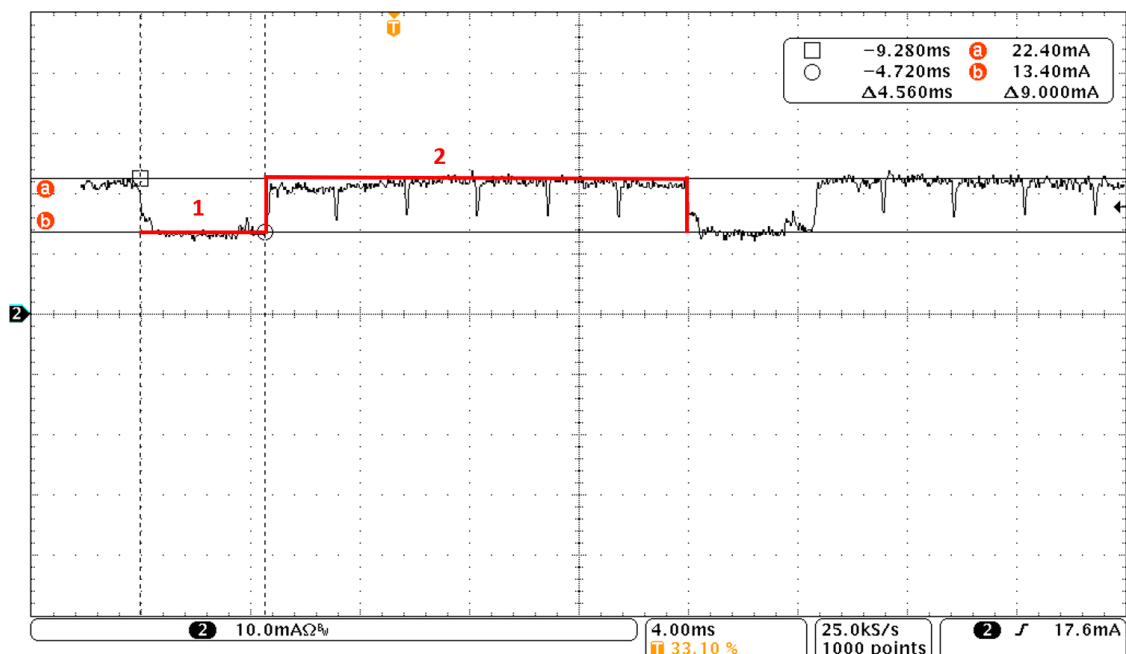
$$I_{AVG} = \frac{1}{T_S} \left(T_{MESH} \cdot \frac{T_1}{T_{12}} \cdot I_1 + T_{MESH} \cdot \frac{T_2}{T_{12}} \cdot I_2 + 16 \cdot T_3 \cdot I_3 + 16 \cdot T_4 \cdot I_4 \right)$$

$$I_{AVG} = \frac{1}{10^4} \left(9870 \cdot \frac{15}{20} \cdot 65.6 + 9870 \cdot \frac{5}{20} \cdot 57.4 + 16 \cdot 3 \cdot 72.2 + 16 \cdot 5.125 \cdot 90.4 \right)$$

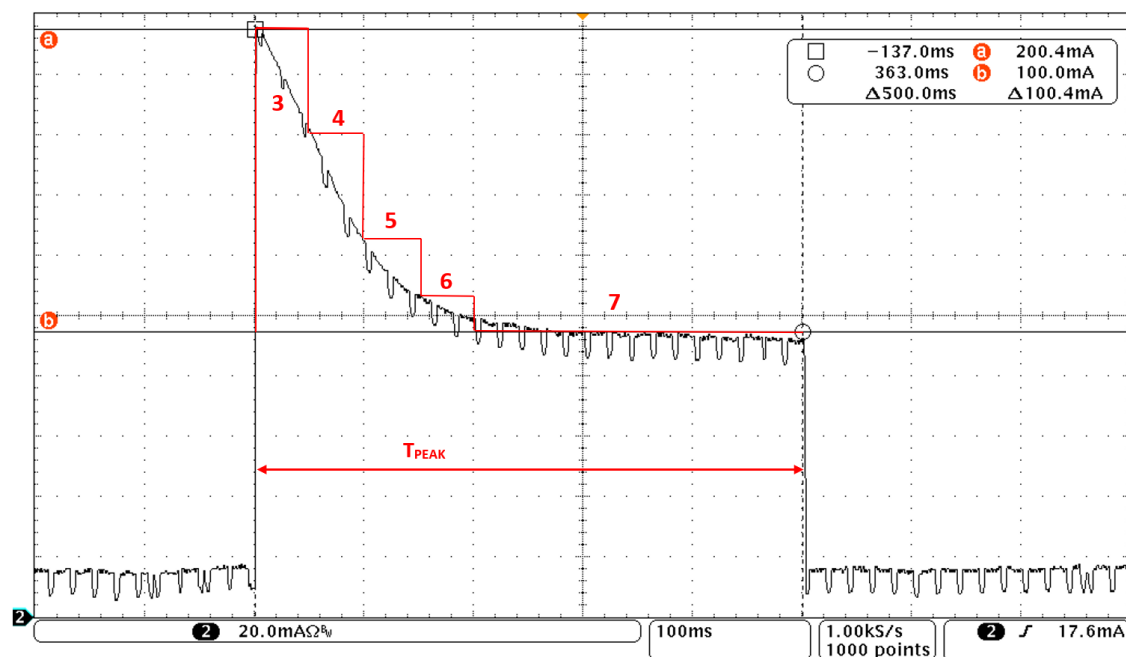
$$\underline{\underline{I_{AVG} = 63.8 \text{ mA}}}$$

7.2.2 Proudový odběr Nucleo68 s MH-Z14A

Deska Nucleo68 byla napájena přes USB USER. Publikovací perioda byla 1 sekunda. Každých 5 sekund dochází ke skokové změně odebíraného proudu. V katalogovém listu snímače CO₂ sice není uveden bližší popis měření koncentrace, nicméně můžeme vyloučit, že by se jednalo o chybu. K proudovému skoku dochází i u snímače MH-Z19B.



Obr. 7.12: Průběh proudu - Nucleo68 se snímačem MH-Z14A



Obr. 7.13: Skoková změna proudu - Nucleo68 se snímačem MH-Z14A

Tab. 7.12: Odběr proudu - Nucleo68 se snímačem MH-Z14A - rozdělení na úseky

Úsek	1	2	3	4	5	6	7
T [ms]	4.56	15.44	50	50	50	50	300
I [mA]	13.4	22.4	200.4	164	128	110	100

Označíme si $T_{MESH} = T_S - T_{PEAK}$, kde $T_S = 5 \text{ sec}$ a $T_{PEAK} = 500 \text{ ms}$. Dále $T_{12} = T_1 + T_2$. Průměrný odběr proudu:

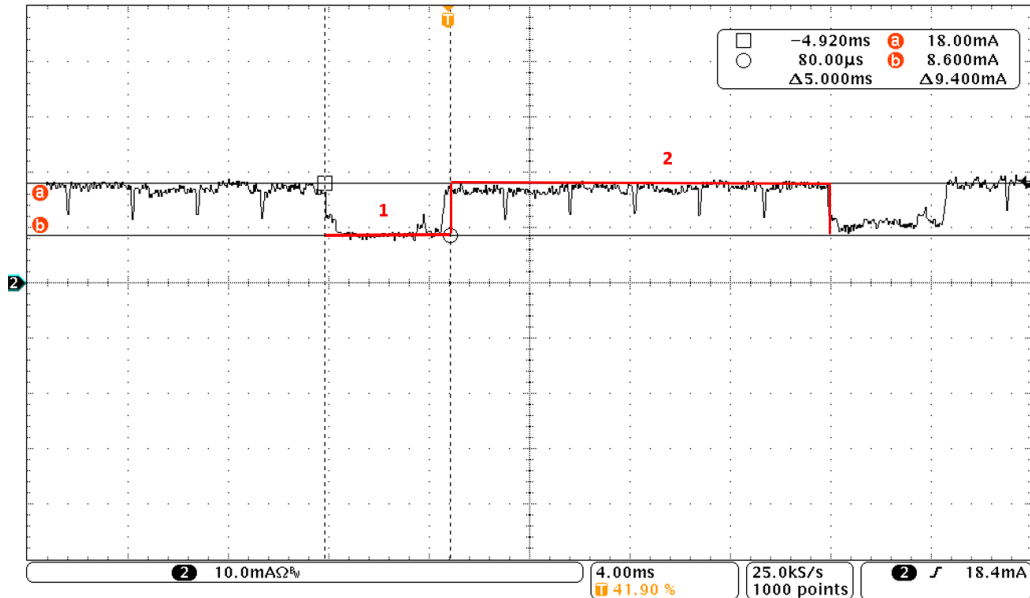
$$I_{AVG} = \frac{1}{T_S} \left(T_{MESH} \cdot \frac{T_1}{T_{12}} \cdot I_1 + T_{MESH} \cdot \frac{T_2}{T_{12}} \cdot I_2 + \sum_{i=3}^7 T_i \cdot I_i \right)$$

$$I_{AVG} = \frac{1}{5000} \left(4500 \cdot \frac{4.56}{20} \cdot 13.4 + 4500 \cdot \frac{15.44}{20} \cdot 22.4 + 60120 \right)$$

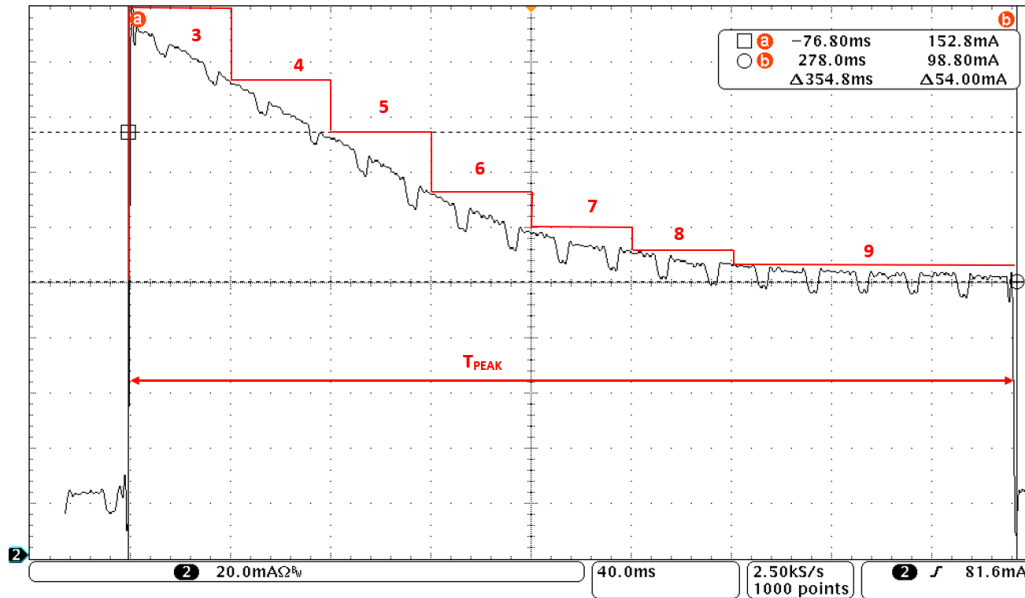
$$\underline{\underline{I_{AVG} = 30.3 \text{ mA}}}$$

7.2.3 Proudový odběr WB55 USB Dongle s MH-Z19B

WB55 USB Dongle byl napájen přes USB USER. Publikovací perioda byla 1 sekunda. Výše již bylo zmíněno, že i při použití MH-Z19B dochází ke skokovým změnám odběru proudu každých 5 sekund.



Obr. 7.14: Průběh proudu - WB55 USB Dongle se snímačem MH-Z19B



Obr. 7.15: Skoková změna proudu - WB55 USB Dongle se snímačem MH-Z19B

Tab. 7.13: Odběr proudu - WB55 USB Dongle se snímačem MH-Z19B - rozdělení na úseky

Úsek	1	2	3	4	5	6	7	8	9
T [ms]	5	15	40	40	40	40	40	40	114.8
I [mA]	8.6	18	198.8	170.8	154.8	130.8	118.8	110.8	98.8

Opět platí $T_{MESH} = T_S - T_{PEAK}$, kde $T_S = 5 \text{ sec}$ a $T_{PEAK} = 354.8 \text{ ms}$. Dále $T_{12} = T_1 + T_2$. Průměrný odběr proudu:

$$I_{AVG} = \frac{1}{T_S} \left(T_{MESH} \cdot \frac{T_1}{T_{12}} \cdot I_1 + T_{MESH} \cdot \frac{T_2}{T_{12}} \cdot I_2 + \sum_{i=3}^9 T_i \cdot I_i \right)$$

$$I_{AVG} = \frac{1}{5000} \left(4500 \cdot \frac{5}{20} \cdot 8.6 + 4500 \cdot \frac{15}{20} \cdot 18 + 40737.04 \right)$$

$$\underline{\underline{I_{AVG} = 23.4 \text{ mA}}}$$

7.3 Měření parametrů LPN

Úspěšnost propagace datové zprávy byla v případě LPN měřena jen pro 2 konfigurace. Při měření byly v síti aktivní pouze LPN, friend a koncentrátor. Payload byl 7 bytů, vysílací výkon 0 dBm ReceiveWindow 255 ms, ReceiveDelay 150 ms, perioda pull requestu low power uzlu 5 sekund a stejná byla i perioda publikování. Každou zprávu odesílal LPN pouze jednou.

Konfigurace 1 (LPN)

Umístění uzlů opět odpovídá rozmístění na obr. 5.1. Při této konfiguraci byl na pozici S1 umístěn LPN, na pozici R1 byl umístěn friend a koncentrátor byl na své pozici C. Úspěšnost doručení z LPN na koncentrátor byla vyhodnocována přibližně ze 100 odeslaných zpráv (viz tabulka 7.14). Průměrná úspěšnost doručení zprávy je při této konfiguraci $(49.7 \pm 7.8)\%$.

Tab. 7.14: Úspěšnost doručení zprávy z LPN (konfigurace 1)

Číslo měření	1	2	3	4	5
Odeslané zprávy	100	101	100	102	101
Přijaté zprávy	60	51	46	38	55
Úspěšnost [%]	60.0	50.5	46.0	37.3	54.5

Konfigurace 2 (LPN)

Při této konfiguraci byly všechny 3 prvky sítě (LPN, friend, koncentrátor) položeny vedle sebe. Z naměřených hodnot byla určena průměrná úspěšnost doručení zprávy $(73.1 \pm 4.1)\%$.

Tab. 7.15: Úspěšnost doručení zprávy z LPN (konfigurace 2)

Číslo měření	1	2	3	4	5
Odeslané zprávy	100	100	100	100	100
Přijaté zprávy	73	70	81	72	70
Úspěšnost [%]	72.3	70.0	81.0	72.0	70.0

Odběr LPN ve spánkovém režimu (WB55 USB Dongle)

Pro měření proudu byl použit multimetr UNI-T UT70B nastavený na rozsah $400 \mu A$, počet digitů $d = 2$.

Tab. 7.16: Měřený odběr proudu ve spánkovém režimu (WB55 USB Dongle)

Číslo měření	1	2	3	4	5	6	7	8	9	10
$I [\mu A]$	96.4	96.2	96.3	96.3	96.2	96.3	96.2	96.2	96.3	96.4

Průměrná hodnota proudu je $96.28 \mu A$, odchylka $0.0748 \mu A$. Nejistota typu A:

$$u_A = \frac{1}{\sqrt{n-1}} \cdot \sigma = \frac{1}{3} \cdot 0.0748 = 0.025 \mu A$$

Nejistota typu B:

$$u_B = \frac{|\Delta_P|}{\sqrt{3}} = \frac{|\delta_M \cdot X_M| + \left| \frac{d}{D} \cdot 100 \cdot X_R \right|}{\sqrt{3}} = \frac{|1 \cdot 96.28| + \left| \frac{2}{3999} \cdot 100 \cdot 400 \right|}{\sqrt{3}} = 0.68 \mu A$$

Kombinovaná standardní nejistota C:

$$u_C = \sqrt{u_A^2 + u_B^2} = \sqrt{0.025^2 + 0.68^2} = 0.68 \mu A$$

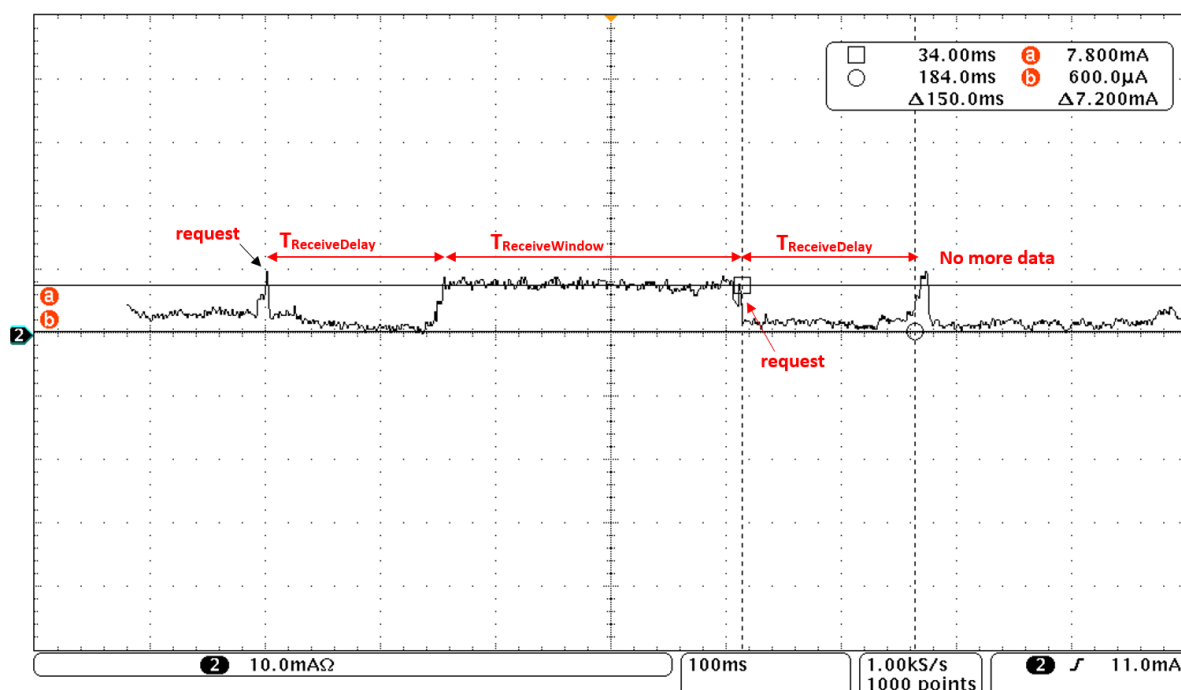
Rozšířená standardní nejistota:

$$U = 2 \cdot u_C = 1.4 \mu A$$

Po výpočtu nejistot měření můžeme konstatovat, že průměrná hodnota odběru proudu je ve spánkovém režimu $(96.3 \pm 1.4) \mu A$ s pravděpodobností 95%.

Komunikace LPN a friend uzlu

V poslední části této kapitoli si ukážeme zaznamenaný průběh proudu, kdy LPN požádá friend uzel o data, přejde do spánku a po následném probuzení přijímá data. LPN měl stejné nastavení parametrů jako v Měření parametrů LPN, pouze nepublikoval. Z obrázku níže si můžeme potvrdit nastavení těchto parametrů. Pro upřesnění doplníme, že LPN může přejít do spánku i před vypršením ReceiveWindow. Záleží na přijatých datech.



Obr. 7.16: Průběh proudu LPN při požadavku na data a jejich čtení

Závěr

Cílem této práce bylo realizovat flexibilní bezdrátový systém měřící koncentraci oxidu uhličitého v okolním prostředí. Nejprve bylo nutné seznámit se se standardem Bluetooth Mesh, ze kterého vycházel návrh architektury měřicího systému. Realizovaný systém je založen na topologii mesh. Pro vytvořený systém byly změřeny tyto parametry: úspěšnost přenosu datové zprávy, délka propagace datové zprávy a průměrný odběr proudu zařízení.

Měřicí systém byl navržen na podmínky uvedené v kapitole Podmínky a prostory pro měřicí systém. Systém sestává z koncentrátoru, který zároveň zajišťuje správu sítě mesh, a z měřicích bodů, které publikují měřené hodnoty koncentrace prostřednictvím Sensor Server modelu. Model Sensor Client implementovaný na koncentrátoru sbírá měřená data a podle požadavku uživatele jsou data ukládána na SD kartu, případně vypisována na obrazovku prostřednictvím sériové linky. Uživatel může za běhu systému přidávat a odebírat zařízení, aniž by musel restartovat celý systém. U všech uzlů sítě kromě koncentrátoru je možné dynamicky měnit adresy pro publikování a odebírání zpráv. Dále byl na uzly měřící koncentraci implementován model Generic OnOff Server, který umožňuje vzdálenou kalibraci připojeného snímače. Vzdálená kalibrace je demonstrována na rozsvěcování a zhašení LED diody.

V rámci práce se nepovedlo uvést do funkčního stavu přímé přistupování k měřicímu zařízení prostřednictvím uzlu proxy. Jelikož jsem byl v podstatě omezen na vývojové moduly STMicroelectronics, neměl jsem možnost otestovat implementaci proxy uzlu od jiných výrobců. Dále se nepovedlo začlenit do procesu provisioning metodu ověření Output OOB.

Úspěšnost doručení zprávy při konfiguraci 1 je z větší části menší než 90% . Přidáním uzlu relay mezi výchozí a cílový uzel měření stoupla úspěšnost na více než 90%, a to ve více než polovině případů. Pokles úspěšnosti můžeme pozorovat při konfiguraci 4, kde byla úspěšnosti přibližně 50%.

Při všech konfiguracích byla při publikovací periodě 800 ms změřena latence propagace datové zprávy menší než 500 ms. Latence zpráv publikovaných každých 100 ms byla vždy větší než 1 sekunda. Ve výsledcích konfigurace 2 můžeme vidět, že latence se nijak výrazně neměnila s různým počtem přeskoků zprávy. Mírné změny můžeme pozorovat při měření konfigurace 4. S vyšším počtem přeskoků by byla změna latence pravděpodobně znatelnější.

Průměrný odběr proudu koncentrátorem při logování s periodou 10 sekund je 63.8 mA. Přibližně polovinu odebírá debugger ST-LINK. V případě měřicího bodu se senzorem je průměrný proudový odběr 30.3 mA. Deska WB55 USB Dongle spolu se senzorem MH-Z19B průměrně odebírá proud 23.4 mA. Průměrný odběr proudu low power uzlu je ve spánkovém režimu $(96.3 \pm 1.4) \mu A$ s pravděpodobností 95%.

Literatura

- [1] AFANEH, Mohammad. *Intro to Bluetooth Low Energy. NovelBits* [online]. 2018 [cit. 2020-12-23]. Dostupné z: <https://www.novelbits.io/introduction-to-bluetooth-low-energy-book/>
- [2] *Bluetooth Core Specification Version 5.0* [online]. Washington: Bluetooth SIG, 2016 [cit. 2020-10-28]. Dostupné z: <https://www.bluetooth.com/specifications/bluetooth-core-specification/>
- [3] *Bluetooth® Low Energy (BLE) Physical Layer. Microchip.com* [online]. wiki-dot.com, 2020 [cit. 2020-10-28]. Dostupné z: <https://microchipdeveloper.com/wireless:ble-phy-layer>
- [4] WOOLLEY, Martin. *Bluetooth Core Specification Version 5.0 Feature Overview* [online]. Washington: Bluetooth SIG, 2016 [cit. 2020-10-28]. Dostupné z: <https://www.bluetooth.com/bluetooth-resources/bluetooth-5-go-faster-go-further/>
- [5] BLAŽEK, Vojtěch. *Sporttester s rozhraním Bluetooth LE* [online]. Brno, 2020, 63 s. [cit. 2020-12-2]. Dostupné z: <https://www.vutbr.cz/studenti/zav-prace/detail/126043>. Diplomová práce. Vysoké učení technické v Brně. Vedoucí práce: Ing. Ondřej Krajsa, Ph.D.
- [6] BUI, Hung. *Bluetooth Core Specification Version 5.0 Feature Overview* [online]. Nordic Semiconductor, 2015 [cit. 2020-12-15]. Dostupné z: <https://devzone.nordicsemi.com/nordic/short-range-guides/b/bluetooth-low-energy/posts/bluetooth-smart-and-the-nordics-softdevices-part-1>
- [7] *Assigned numbers and GAP* [online]. Washington: Bluetooth SIG, 2020 [cit. 2020-12-21]. Dostupné z: <https://www.bluetooth.com/specifications/assigned-numbers/generic-access-profile/>
- [8] *16-bit UUID Numbers* [online]. Bluetooth SIG, 2020 [cit. 2020-12-23]. Dostupné z: <https://btprodspecificationrefs.blob.core.windows.net/assigned-values/16-bit%20UUID%20Numbers%20Document.pdf>
- [9] ANDERSON, Mike. *Understanding Bluetooth Mesh Networking* [online]. Tech Briefs Media Group, 2019 [cit. 2020-12-28]. Dostupné z: <https://www.techbriefs.com/component/content/article/tb/supplements/st/features/articles/33885>

- [10] *Bluetooth mesh addressing modes* [online]. O'Reilly Media, Inc., 2020 [cit. 2020-12-28]. Dostupné z: <https://www.oreilly.com/library/view/internet-of-things/9781788470599/56b7d7c4-f49f-44f7-bd11-baac07868db1.xhtml>
- [11] ČESKO. Nařízení vlády č. 361/2007 Sb., nařízení vlády, kterým se stanoví podmínky ochrany zdraví při práci. In: *Zákony pro lidi.cz*, [online]. © AION CS 2010-2020 [cit. 30. 12. 2020]. Dostupné z: <https://www.zakonyprolidi.cz/cs/2007-361>
- [12] VAFEK, Zdeněk. *Možnosti měření oxidu uhličitého – měřicí přístroje a čidla* [online]. TZB-info, 2016 [cit. 2020-12-30]. Dostupné z: <https://vetrani.tzb-info.cz/vnitřni-prostredi/13910-moznosti-mereni-oxidu-uhliciteho-merici-pristroje-a-cidla>
- [13] *Questions and Answers about Axetris Infrared Sources* [online]. Axetris AG, 2019 [cit. 2020-12-30]. Dostupné z: <https://www.axetris.com/en/irs/frequently-asked-questions/frequently-asked-questions>
- [14] BÁRTA, Aleš. *Řízení kvality vzduchu v budovách (koncentrace CO₂)* [online]. Pardubice, 2018, 69 s. [cit. 2020-12-31]. Dostupné z: https://dk.upce.cz/bitstream/handle/10195/70723/BartaA_RizeniKvality_DH_2018.pdf?sequence=1&isAllowed=y. Diplomová práce. Univerzita Pardubice. Vedoucí práce: Ing. Daniel Honc, Ph.D.
- [15] *STM32CubeWB* [online]. STMicroelectronics, 2020 [cit. 2020-12-30]. Dostupné z: <https://www.st.com/en/embedded-software/stm32cubewb.html>
- [16] *Getting started with nRF5 SDK and SES* [online]. Nordic Semiconductor, 2020 [cit. 2021-01-02]. Dostupné z: https://infocenter.nordicsemi.com/topic/ug_gsg_ses/UG/gsg/intro.html
- [17] *SoftDevices* [online]. Nordic Semiconductor, 2020 [cit. 2021-01-02]. Dostupné z: https://infocenter.nordicsemi.com/topic/ug_gsg_ses/UG/gsg/softdevices.html
- [18] *STM32WBx5* [online]. STMicroelectronics, 2020 [cit. 2021-01-02]. Dostupné z: <https://www.st.com/en/microcontrollers-microprocessors/stm32wbx5.html#documentation>
- [19] *Mesh Profile* [online]. Washington: Bluetooth SIG, 2019 [cit. 2021-02-10]. Dostupné z: <https://www.bluetooth.com/specifications/specs/mesh-profile-1-0-1/>

- [20] *Mesh Model* [online]. Washington: Bluetooth SIG, 2019 [cit. 2021-02-10]. Dostupné z: <https://www.bluetooth.com/specifications/specs/mesh-mode-1-1-0-1/>
- [21] *Mesh Device Properties* [online]. Washington: Bluetooth SIG, 2020 [cit. 2021-02-10]. Dostupné z: <https://www.bluetooth.com/specifications/specs/mesh-device-properties-2/>
- [22] *How to build a Bluetooth® Low Energy mesh application for STM32WBx5 line microcontrollers* [online]. STMicroelectronics, 15.12.2020 [cit. 2021-02-13]. Dostupné z: https://www.st.com/resource/en/application_note/dm00600575-how-to-build-a-bluetooth-low-energy-mesh-application-for-stm32wbx5-microcontrollers-stmicroelectronics.pdf
- [23] *nRF5 SDK for Mesh v5.0.0* [online]. Nordic Semiconductor, 2021 [cit. 2021-03-16]. Dostupné z: https://infocenter.nordicsemi.com/topic/struct_sdk/struct/sdk_mesh_latest.html
- [24] *nRF5 SDK v17.0.2* [online]. Nordic Semiconductor, 2021 [cit. 2021-03-16]. Dostupné z: https://infocenter.nordicsemi.com/topic/struct_sdk/struct/sdk_nrf5_latest.html
- [25] *GATT Specification Supplement* [online]. Washington: Bluetooth SIG, 2021 [cit. 2021-03-20]. Dostupné z: <https://www.bluetooth.com/specifications/specs/gatt-specification-supplement-4/>
- [26] *BLE Mesh Security Overview* [online]. Laird Connectivity, 2018 [cit. 2021-03-21]. Dostupné z: <https://connectivity-staging.s3.us-east-2.amazonaws.com/s3fs-public/2018-10/BLE%20Mesh%20Security%20Overview.pdf>
- [27] E. D. Leon and M. Nabi, *An Experimental Performance Evaluation of Bluetooth Mesh Technology for Monitoring Applications* [online]. 2020 IEEE Wireless Communications and Networking Conference (WCNC), 2020, pp. 1-6 [cit. 2021-03-29]. Dostupné z: <https://ieeexplore.ieee.org/document/9120762>
- [28] *Infrared CO2 Sensor Module MH-Z19B* [online]. Zhengzhou: Winsen Electronics Technology Co., 23.9.2019 [cit. 2021-04-01]. Dostupné z: <https://www.winsen-sensor.com/d/files/MH-Z19B.pdf>
- [29] *Intelligent Infrared Carbon Dioxide Module MH-Z14A* [online]. Zhengzhou: Winsen Electronics Technology Co., 5.5.2019 [cit. 2021-04-01]. Dostupné z: <https://www.winsen-sensor.com/d/files/MH-Z14A.pdf>

- [30] THORSRUD, Einar. *nRF52840 Dongle Programming Tutorial* [online]. Nordic Semiconductor, 27.11.2018 [cit. 2021-04-06]. Dostupné z: <https://devzone.nordicsemi.com/nordic/short-range-guides/b/getting-started/posts/nrf52840-dongle-programming-tutorial>
- [31] Darroudi, S.M.; Caldera-Sánchez, R.; Gomez, C. *C. Bluetooth Mesh Energy Consumption: A Model*. Sensors 2019, 19, 1238. [cit. 2021-04-25]. Dostupné z: <https://doi.org/10.3390/s19051238>

Seznam příloh

A	Projekty a zdrojové kódy	82
A.1	Koncentrátor	82
A.2	Nucleo68 - Sensor server	82
A.3	WB55 USB Dongle - Sensor server	82

A Projekty a zdrojové kódy

Pro elektronickou verzi se tato příloha nachází v přiloženém komprimovaném souboru. V tištěné verzi se příloha nachází na přiloženém CD.

A.1 Koncentrátor

Projekt pro koncentrátor se nachází v adresáři `..\mesh_koncentrator\Projects\P-NUCLEO-WB55.Nucleo\Applications\BLE\BLE_MeshLightingProvisioner\STM32CubeIDE`.

A.2 Nucleo68 - Sensor server

Projekt pro měřicí bod implementovaný na Nucleo68 se nachází v adresáři `..\mesh_dk_sensor\Projects\P-NUCLEO-WB55.Nucleo\Applications\BLE\BLE_MeshLightingPRFNode\STM32CubeIDE`. Před zahájením překlady programu je nutné překopírovat knihovny *libBle_Mesh_CM4_GCC.a* a *libBle_Mesh_CM4_GCC_DEBUG.a* z adresáře `..\mesh_koncentrator\Middlewares\ST\STM32_WPAN\ble\mesh\Library` do adresáře `..\mesh_dk_sensor\Middlewares\ST\STM32_WPAN\ble\mesh\Library`.

A.3 WB55 USB Dongle - Sensor server

Projekt pro měřicí bod implementovaný na WB55 USB Dongle se nachází v adresáři `..\mesh_dongle_sensor\Projects\P-NUCLEO-WB55.USB Dongle\Applications\BLE\BLE_MeshLightingPRFNode\STM32CubeIDE`. Před zahájením překlady programu je nutné překopírovat knihovny *libBle_Mesh_CM4_GCC.a* a *libBle_Mesh_CM4_GCC_DEBUG.a* z adresáře `..\mesh_koncentrator\Middlewares\ST\STM32_WPAN\ble\mesh\Library` do adresáře `..\mesh_dongle_sensor\Middlewares\ST\STM32_WPAN\ble\mesh\Library`. Adresář `..\mesh_dongle_sensor\Projects\P-NUCLEO-WB55.USB Dongle\Applications` obsahuje i projekt pro LPN.